HP Process Automation

Process Studio™ User Guide

Version 7.3
Document Revision 0
01 May 2012

Connectis.ca
DOCUMENT SOLUTIONS
905.695.2200
888.707.8221
info@connectis.ca
www.connectis.ca

Process Automation®

Power

Notice

This documentation is a proprietary product of Autonomy and is protected by copyright laws and international treaty. Information in this documentation is subject to change without notice and does not represent a commitment on the part of Autonomy. While reasonable efforts have been made to ensure the accuracy of the information contained herein, Autonomy assumes no liability for errors or omissions. No liability is assumed for direct, incidental, or consequential damages resulting from the use of the information contained in this documentation.

The copyrighted software that accompanies this documentation is licensed to the End User for use only in strict accordance with the End User License Agreement, which the Licensee should read carefully before commencing use of the software. No part of this publication may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of the copyright owner.

This documentation may use fictitious names for purposes of demonstration; references to actual persons, companies, or organizations are strictly coincidental.

Trademarks and Copyrights

Copyright 2012 Autonomy Corporation plc and all its affiliates. All rights reserved. ACI API, Alfresco Connector, Arcpliance, Autonomy Process Automation, Autonomy Fetch for Siebel eBusiness Applications, Autonomy, Business Objects Connector, Cognos Connector, Confluence Connector, ControlPoint, DAH, Digital Safe Connector, DIH, DiSH, DLH, Documentum Connector, DOH, EAS Connector, Ektron Connector, Enterprise AWE, eRoom Connector, Exchange Connector, FatWire Connector, File System Connector for Netware, File System Connector, FileNet Connector, FileNet P8 Connector, FTP Fetch, HTTP Connector, Hummingbird DM Connector, IAS, IBM Content Manager Connector, IBM Seedlist Connector, IBM Workplace Fetch, IDOL Server, IDOL, IDOLme, iManage Fetch, IMAP Connector, Import Module, iPlanet Connector, KeyView, KVS Connector, Legato Connector, LiquidOffice, LiquidPDF, LiveLink Web Content Management Connector, MCMS Connector, MediClaim, Meridio Connector, Meridio, Moreover Fetch, NNTP Connector, Notes Connector, Objective Connector, OCS Connector, ODBC Connector, Omni Fetch SDK, Open Text Connector, Oracle Connector, PCDocs Fetch, PLC Connector, POP3 Fetch, Portal-in-a-Box, RecoFlex, Retina, SAP Fetch, Schlumberger Fetch, SharePoint 2003 Connector, SharePoint 2007 Connector, SharePoint 2010 Connector, SharePoint Fetch, SpeechPlugin, Stellent Fetch, TeleForm, Tri-CR, Ultraseek, Verity Profiler, Verity, VersiForm, WebDAV Connector, WorkSite Connector, and all related titles and logos are trademarks of Autonomy Corporation plc and its affiliates, which may be registered in certain jurisdictions.

Microsoft is a registered trademark, and MS-DOS, Windows, Windows 95, Windows NT, SharePoint, and other Microsoft products referenced herein are trademarks of Microsoft Corporation.

UNIX is a registered trademark of The Open Group.

AvantGo is a trademark of AvantGo, Inc.

Epicentric Foundation Server is a trademark of Epicentric, Inc.

Documentum and eRoom are trademarks of Documentum, a division of EMC Corp.

FileNet is a trademark of FileNet Corporation.

Lotus Notes is a trademark of Lotus Development Corporation.

mySAP Enterprise Portal is a trademark of SAP AG.

Oracle is a trademark of Oracle Corporation.

Adobe is a trademark of Adobe Systems Incorporated.

Novell is a trademark of Novell, Inc.

Stellent is a trademark of Stellent, Inc.

All other trademarks are the property of their respective owners.

Notice to Government End Users

If this product is acquired under the terms of a **DoD contract:** Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of 252.227-7013. **Civilian agency contract:** Use, reproduction or disclosure is subject to 52.227-19 (a) through (d) and restrictions set forth in the accompanying end user agreement. Unpublished-rights reserved under the copyright laws of the United States. Autonomy, Inc., One Market Plaza, Spear Tower, Suite 1900, San Francisco, CA. 94105, US.

01 May 2012

Contents

About This Document 13
Documentation Updates 13
Related Documentation 14
Conventions 15
Notational Conventions 15
Command-line Syntax Conventions 16
Notices 17
Autonomy Product References 17
Autonomy Customer Support 18
Contact Autonomy 18

Part 1 Getting Started

Chapter 1
Introduction 21
Autonomy Process Automation 21
Autonomy Process Automation Components 22

Chapter 2
Process Studio Interface and Commands 25
Process Studio User Interface 26
Templates and Instances 27
Graphical View 27
XML View 27
Tooltips 28
Customize Process Studio 28
Customize the Default Scripting Language 28
Customize the Task Library 28

Customize the Toolbar 29
Process Studio Commands 29
Open a New Window 30
Zoom Controls 30
Undo or Redo an Operation 31
Show or Hide Grid Lines 31
Compact Tasks 31
View Messages 31
Exit Process Studio 32
Build Expressions 32
Create Dynamic Values Using Substitution Syntax 34
Substitution Syntax Definitions 35
Additional Information 37
Create Custom Scripts 37
Check a Script for Syntax Errors 39
Set the Script Language in the Script Editor 39
Find and Replace Text in the Script Editor 39
Insert a Function into a Script 40
Insert a Task Reference into a Script 40

Part 2 Create and Deploy Processes

Chapter 3
Create and Save a Process 45
Create a Process 45
Create a New Process 46
Open a Process 46
Retrieve a Process from the APA Server 47
Save a Process 48
Save As 48
Save a Process as an Image 49
Print a Process 49
Close a Process 50

Chapter 4
Configure a Process 51
Configure Process Properties 51
General Process Properties 52
Configure Attributes 53

Add an attribute 53
Delete an attribute 54
View and Configure Process Fields 54
Add a Custom Field to the Process Definition 54
Delete a Field from the Process Definition 54
Data Types for Process Fields 54
Configure Initial Form Properties 56
Form Packets 57
Create and Configure a New Form Packet 57
Delete a Form Packet 59
Form Packet Rules 59
Create a Rule 61
Configure the Set Node State Consequence 63
Configure the Set Message in TOC Consequence 64
Configure the Add Form Consequence 65
Configure the Duplicate Node Consequence 66
Configure the Remove Nodes Consequence 67
Configure the Chain to Specific Node Consequence 68

Chapter 5
Tasks Overview 69
Tasks Overview 69
Process Control Tasks 72
Container Task 72
Sub-process Task 72
And Task 72
Or Task 73
Branch Task 74
Loop Task 74
Gate Task 75
Delay Task 75
Escalate Task 75
Exception Task 76
Form Tasks 77
Form Task 77
Form Copy Task 77
Form Export Task 77
Form Packet Task 77
Utility Tasks 77

Script Task 77
Command Task 78
Message Task 78
E-mail Task 78
Manual Task 78
Web Service Task 79
Other Tasks 79
Composite Task 79
Sticky Note Task 80

Chapter 6
Add and Connect Tasks 81
Add a Task 81
Add a Task Pattern 82
Add a Container Task 82
Add Tasks to a Container 82
Connect Tasks 83
Connect a Branch Task 84
Connect a Loop Task 85
Connect an Exception Task 85
Connect an Escalate Task 86
Remove a Connection 87
Save a Composite Task 87
Delete a Task 88
Convert a Task to another Type 88
Add a Comment to a Process 89

Chapter 7
Configure Tasks 91
Configure Tasks 91
Configure General Properties 92
Name 93
Duration 93
Scoped ID 93
Design Comments 94
Enable Script 94
Configure Attributes 94
Add an Attribute 94
Delete an Attribute 95
Task Assignment 95

Assignment Priority 96
Abort if Unassigned 97
Configure Task Assignment 97
Configure Task-specific Properties 98
Configure a Sub Process Task 98
Configure an OR Task 99
Configure a Branch Task 100
Use Another Task to Determine Path Selection 101
Configure a Loop Task 102
Use Another Task to Control Loop Execution 103
Configure a Delay Task 103
Configure an Escalate Task 104
Configure a Script Task 104
Configure a Command Task 105
Configure a Message Task 105
Configure a Committee Task 106
Configure an E-mail Task 107
Configure a Manual Task 107
Configure a Web Service Task 108
Configure a Form Task 113
Configure a Form Copy Task 114
Configure a Form Export Task 114
Configure a Form Packet Task 114
Task Properties 116
Use Business Calendar 117
Allow Form to be Edited 117
Set Field Properties 117
Allow Attachments to be Deleted 118
Initialize Form at this Task 118
Customize Submit Action List 118
Default Submit Actions 119
Enable Form on Mobile devices 119
Configure the form fields to display on mobile devices 120
Allow Viewing of Progress Map 121
Display Page for Entering Subject and Notes 121
Automatically Display Next Form on Submit 121
Represent Static Table Cells as Top Level Fields 121

Chapter 8
Simulate a Process 123
Flow Simulation 123
Create a Flow Simulation 123
Duplicate an Instance 124
Delete an Instance 124
Task states 125
Flow Simulation Commands 125
Change the State of Aborted Tasks 126
Cancel a Task 126
Open a Gate Task 126
Load Simulation 127
Configure Tasks for Load Simulation 127
Configure Simulation Properties for Loop Tasks 128
Configure Simulation Properties for Branch Tasks 128
Create and Run a Load Simulation 129

Chapter 9
Connect to an APA Server 131
Connect to the APA Server 131
Disconnect from the APA Server 132
Search the APA Server 132

Chapter 10
Publish a Process 135
Publishing Overview 135
Publishing and Permissions 136
Publish to Multiple Servers 136
Publish a Process 137
Publishing Wizard: General Properties 138
Publishing Wizard: Permissions 141
Configure Permissions 141
Roles, Groups, and Users 142
Add a user or LDAP group 142
Add a Role 143
Permissions 143
Publishing Wizard: Process Settings 145
Initiate a Process 146
Initiate a Process from the Web Desktop 146
Other Methods to Initiate a Process 147

Order of Initialization 148

Part 3 Process Studio Reference

Chapter 11
Task Patterns 151
Simple Design Patterns 151
Sequence 151
Parallelism 152
Branch 153
Loop 154
Join 154
And 155
Or 156
Escalate 156
Exception 157
Task Patterns 157
Sequence 157
Parallel Split 158
Synchronization 158
Exclusive Choice 159
Multi-choice 159
Simple Merge 160
Synchronizing Merge 161
Multi-merge 161
Discriminator 162
Arbitrary Cycles 163
Implicit Termination 163
Multiple Instances without Synchronization 164
Multiple Instances with a Priori Design Knowledge 164
Multiple Instances with a Priori Runtime Knowledge 165
Multiple Instances without a Priori Runtime Knowledge 166
Deferred Choice 166
Interleaved Parallel Routing 167
Milestone 168
Cancel Activity 169
Cancel Case 170

Chapter 12
APA Expression Language 171
Supported Functions and Operators 172
Expression Language Data Types 172
String Data Type 173
Number Data Type 173
Date Data Type 174
Time Data Type 174
Boolean Data Type 175
Field References 175
Literal Values 176
Wildcard References 176
Expression Language Operators 177
Grouping Operator 177
Logical Operators 177
Arithmetic Operators 178
Multiplicative Operators 178
Additive Operators 178
Comparison Operators 179
Equality Operators 179
Relational Operators 179
Precedence 179
Evaluation 180
Conversion 180
Selection Operator 180
String Concatenation Operator 180
Expression Language Functions 180
Boolean Functions 181
Date Functions 181
Time Functions 181
String Functions 182
Numeric Functions 183
Task Functions 184

Part 4 Process Studio Tutorial

Chapter 13
Getting Started 187
Tutorial Overview 188
Tutorial Prerequisites 188
Create Relationships and Work Queues 189
Add Relationships 189
Add a Work Queue 190
Publish the Purchase Order Form 191

Chapter 14
Process Studio Tutorial 193
Create a Process Template and Add Tasks 194
Create a New Process Template 194
Add a Form Task to the Template 195
Configure the Form Task 195
Save the Process Template 198
Create the Supervisor Review 199
Add a Second Form Task to the Process Template 199
Configure the Second Form Task 200
Assign the Form Task Using a Relationship 204
Add a Branch Task and Successor Tasks 205
Add a Branch Task to the Process 206
Add Successor Tasks to the Branch Task 206
Configure the Branch Task 207
Configure the Loop Task 210
Configure the Loop task 210
Define the Approved Path Using an Expression 212
Configure the second Branch Task 212
Configure the Rejected Path 216
Configure the Container Task 216
Configure the Form Copy Task 218
Configure the E-mail Task using Substitution Syntax 220
Configure the CFO Review Task 223
Configure the CFO Review Form Task 223
Configure the Accounting Work Queue Review Task 227
Configure the Accounting WQ Review Form Task 228
Configure the CFO Review Branch Task 232

Add and Connect the CFO Review Branch Task 232
Configure the CFO Review Branch Task 233
Configure the Accounting Review Branch Task 235
Add and Connect the Accounting Review Branch Task 236
Configure the Accounting Review Branch Task 237
Configure the Rejected2 Container 238
Configure the Rejected2 Container Task 239
Add Tasks to the Container 239
Send a Copy of the Form to the Supervisor 240
Configure the Approved Container 244
Configure the Approved Container 244

Index 245

About This Document

This document is for users who need to create processes with Autonomy Process Automation (APA) Process Studio.

- Documentation Updates
- Related Documentation
- Conventions
- Autonomy Product References
- Autonomy Customer Support
- Contact Autonomy

Documentation Updates

The information in this document is current as of Process Studio version 7.3. The content was last modified 01 May 2012.

You can retrieve the most current product documentation from Autonomy's Knowledge Base on the Customer Support Site.

A document in the Knowledge Base displays a *version number* in its name, such as *IDOL Server 7.5 Administration Guide*. The version number applies to the product that the document describes. The document may also have a *revision number* in its name, such as *IDOL Server 7.5 Administration Guide Revision 6*. The revision number applies to the document and indicates that there were revisions to the document since its original release.

It is recommended that you periodically check the Knowledge Base for revisions to documents for the products your enterprise is using.

To access Autonomy documentation

1. Go to the Autonomy Customer Support site at

   https://customers.autonomy.com

2. Click **Login**.
3. Enter the login credentials that were given to you, and then click **Submit**.

   The Knowledge Base Search page opens.

4. In the **Search** box, type a search term or phrase. To browse the Knowledge Base using a navigation tree only, leave the **Search** box empty.
5. Ensure the **Documentation** check box is selected.
6. Click **Search**.

   Documents that match the query display in a results list.

7. To refine the results list, select one or more of the categories in the **Filter By** pane. You can restrict results by
   - **Product Group**. Filters the list by product suite or division. For example, you could retrieve documents related to the iManage, IDOL, Virage or KeyView product suites.
   - **Product**. Filters the list by product. For example, you could retrieve documents related to IDOL Server, Virage Videologger, or KeyView Filter.
   - **Component**. Filters the list by a product's components. For example, you could retrieve documents related to the Content or Category component in IDOL.
   - **Version**. Filters the list by product or component version number.
   - **Type**. Filters the list by document format. For example, you could retrieve documents in PDF or HTML format. Guides are typically provided in both PDF and HTML format.
8. To open a document, click its title in the results list.

   To download a PDF version of a guide, open the PDF version, click the Save icon in the PDF reader, and save the PDF to another location.

Related Documentation

The following documents provide more details on Process Studio:

- *Autonomy Process Automation Administration Guide*

Conventions

The following conventions are used in this document.

Notational Conventions

This document uses the following conventions.

Convention	Usage
Bold	User-interface elements such as a menu item or button. For example: Click **Cancel** to halt the operation.
Italics	Document titles and new terms. For example: ■ For more information, see the *IDOL Server Administration Guide*. ■ An *action command* is a request, such as a query or indexing instruction, sent to IDOL Server.
`monospace font`	File names, paths, and code. For example: The `FileSystemConnector.cfg` file is installed in `C:\Program Files\FileSystemConnector\`.
`monospace bold`	Data typed by the user. For example: ■ Type **`run`** at the command prompt. ■ In the **User Name** field, type **`Admin`**.
`monospace italics`	Replaceable strings in file paths and code. For example: `user` *`UserName`*

Command-line Syntax Conventions

This document uses the following command-line syntax conventions.

Convention	Usage
`[ optional ]`	Brackets describe optional syntax. For example: `[ -create ]`
`|`	Bars indicate “either | or” choices. For example: `[ option1 ] | [ option2 ]` In this example, you must choose between `option1` and `option2`.
`{ required }`	Braces describe required syntax in which you have a choice and that at least one choice is required. For example: `{ [ option1 ] [ option2 ] }` In this example, you must choose `option1`, `option2`, or both options.
`required`	Absence of braces or brackets indicates required syntax in which there is no choice; you must type the required syntax element.
`variable` `<variable>`	Italics specify items to be replaced by actual values. For example: `-merge` *`filename1`* (In some documents, angle brackets are used to denote these items.)
`...`	Ellipses indicate repetition of the same pattern. For example: `-merge` *`filename1`*`,` *`filename2`* `[,` *`filename3`* `... ]` where the ellipses specify, *`filename4`*, and so on.

The use of punctuation—such as single and double quotes, commas, periods—indicates actual syntax; it is not part of the syntax definition.

Notices

This document uses the following notices:

CAUTION A caution indicates an action can result in the loss of data.

IMPORTANT An important note provides information that is essential to completing a task.

NOTE A note provides information that emphasizes or supplements important points of the main text. A note supplies information that may apply only in special cases—for example, memory limitations, equipment configurations, or details that apply to specific versions of the software.

TIP A tip provides additional information that makes a task easier or more productive.

Autonomy Product References

This document references the following Autonomy products:

- Autonomy Process Automation (APA) Portal.
- Autonomy Process Automation (APA) Web Desktop.
- Autonomy Process Automation (APA) Form Designer.
- Autonomy Process Automation (APA) Management Console.

Autonomy Customer Support

Autonomy Customer Support provides prompt and accurate support to help you quickly and effectively resolve any issue you may encounter while using Autonomy products. Support services include access to the Customer Support Site (CSS) for online answers, expertise-based service by Autonomy support engineers, and software maintenance to ensure you have the most up-to-date technology.

To access the Customer Support Site, go to

https://customers.autonomy.com

The Customer Support Site includes:

- **Knowledge Base**: The CSS contains an extensive library of end user documentation, FAQs, and technical articles that is easy to navigate and search.
- **Case Center**: The Case Center is a central location to create, monitor, and manage all your cases that are open with technical support.
- **Download Center**: Products and product updates can be downloaded and requested from the Download Center.
- **Resource Center**: Other helpful resources appropriate for your product.

To contact Autonomy Customer Support by e-mail or phone, go to

http://www.autonomy.com/content/Services/Support/index.en.html

Contact Autonomy

For general information about Autonomy, contact one of the following locations:

Europe and Worldwide	North and South America
E-mail: autonomy@autonomy.com	E-mail: autonomy@autonomy.com
Telephone: +44 (0) 1223 448 000 Fax: +44 (0) 1223 448 001	Telephone: 1 415 243 9955 Fax: 1 415 243 9984
Autonomy Corporation plc Cambridge Business Park Cowley Rd Cambridge CB4 0WZ United Kingdom	Autonomy, Inc. One Market Plaza Spear Tower, Suite 1900 San Francisco CA 94105 USA

PART 1 Getting Started

This section introduces Autonomy Process Automation (APA), and describes how to navigate the APA Process Studio user interface.

- Introduction
- Process Studio Interface and Commands

CHAPTER 1

Introduction

This section introduces Autonomy Process Automation (APA), and describes how APA Process Studio is used in process automation.

- Autonomy Process Automation
- Autonomy Process Automation Components

Autonomy Process Automation

Autonomy Process Automation™ (APA) is a powerful, web-based solution for creating, routing and managing electronic forms. APA is the fastest way for organizations to put forms online and automate business processes.

APA automatically routes forms to the appropriate person or department and notifies managers by e-mail when digital approvals are needed. This results in efficient and secure processing of enterprise information.

APA can fully automate the processes of complex business environments and is the only business process management solution that offers enterprise-class search.

Use Industry Standards

APA uses industry standard formats such as XML, HTML, and PDF.

By using standard formats, organizations can deploy automated online forms without using proprietary form formats and filler applications. The APA SOAP API enables integration with other enterprise application environments.

Enforce Compliance and Best Practices

Process Automation enforces adherence to best practices, corporate policies and government regulations by centralizing process control. It also provides visibility into processes, notifies risk managers of non-compliance and automates corrective action and responses. APA provides a secure audit trail by automatically tracking approvals, reviews, edits and business activity at every step of a process.

Securely Connect Across Your Business

APA securely connects all people across the enterprise, whether inside or outside the enterprise firewall. As a result, all users can stay better informed and work together more effectively.

Advanced form routing significantly reduces paper shuffling as forms are circulated electronically. Role-based routing routes forms based upon a job function or relationship rather than to specific groups or individuals.

APA can integrate with existing LDAP directory services, synchronizing users and groups with a central directory server to provide users with a single login access to the APA server.

Scale Your Processes to Match Your Growing Business

Built to address the complexities of any size organization, APA has a highly scalable, fault-tolerant, open architecture that allows you to accommodate the growth of people, processes and information across your enterprise.

Autonomy Process Automation Components

Autonomy Process Automation includes the following components:

- Form Designer
- Process Studio
- APA Server
- Portal and Web Desktop
- My Data Client

- Management Console
- Optional Components

Form Designer

Form Designer is used to create forms and publish them to the APA server.

Process Studio

Process Studio is used to create process workflows and publish them to the APA server.

APA Server

The APA server hosts the APA client applications, published forms, and processes. The server presents forms to users, and routes the forms on submission. It also handles the initiation, management, and tracking of workflow processes. The APA server enforces the business rules specified by process definitions.

The APA Server can connect to database servers for form validation and data export. The APA Server can also connect to an LDAP server to synchronize users and groups with a central directory.

Portal and Web Desktop

Portal and Web Desktop provide access to published forms and processes, and supply the user interface for routing these forms and processes.

My Data Client

The My Data Client provides users with direct access to data exported from forms. Data can be retrieved from the APA server in a number of formats. These formats are compatible with applications such as Microsoft Excel and Microsoft Access.

Management Console

The Management Console provides a graphical user interface for administrators to manage the APA server.

Optional Components

The following components might also be included with your Autonomy Process Automation solution.

NOTE Many of these items are sold separately and might not be included with your version of APA. Contact your APA administrator for more information.

- **Public Access Option**: The Public Access Option provides access to APA for public users who do not have accounts on the APA Server.
- **Email Routing Option**: The Email Routing Option allows APA users and Anonymous users to route forms to e-mail addresses.
- **Offline Forms Option**: The Offline Forms Option allows users to save forms offline (that is, locally on a hard drive). This is useful when they would like to save a form (perhaps already partially filled out), complete it offline, and then load the form while online for submission to the APA Server.
- **Enterprise Access Option**: The Enterprise Access Option allows users who do not need to approve forms and processes (such as non-managerial employees) to use APA. These users can initiate forms and processes and are provided with log on credentials and user profiles in APA.
- **Mobile Inbox Option**: The Mobile Inbox Option provides APA users who are granted the right to approve forms and processes the ability to approve using a Blackberry mobile device. These users are provided with access to a secure APA Inbox within the Blackberry user interface.
- **Case Management Option:** Autonomy Case Management (ACM) provides comprehensive tools to build sophisticated, nested, dynamic rules allowing end-to-end automation to occur with predetermined compliance and full transparency.

CHAPTER 2

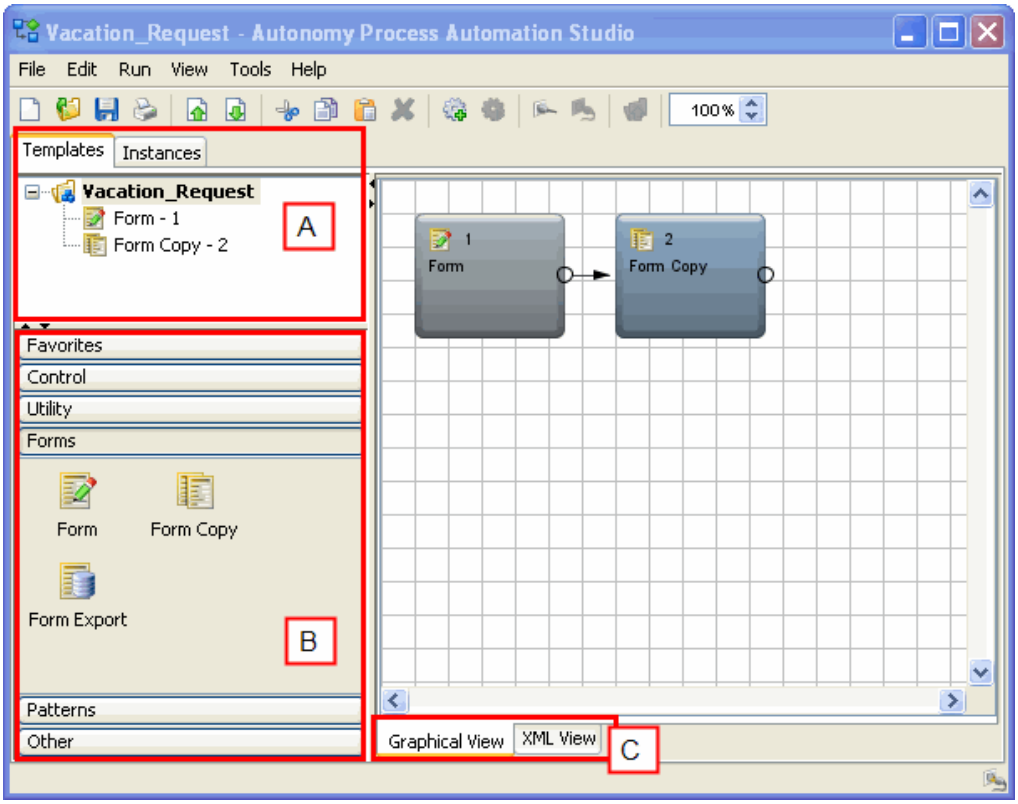
Process Studio Interface and Commands

A *Process* is a set of tasks that are connected together in order to achieve a specific goal. Process Studio provides a graphical user interface for you to create processes and publish them to the APA Server.

This section provides an overview of the Process Studio user interface and the commands that are available for designing processes.

- Process Studio User Interface
- Customize Process Studio
- Process Studio Commands
- Build Expressions
- Create Dynamic Values Using Substitution Syntax
- Create Custom Scripts

Process Studio User Interface



- A. The **Process Tree** displays open process templates and instances, and their associated tasks. See "Templates and Instances" on page 27.

  To view an open process template or instance, double-click its name.

  Click to collapse or expand the **Process Tree**.

  - The **Templates** tab displays open process templates. Root-level nodes represent open processes and display the process title.
  - The **Instances** tab displays instances that have been created in Process Studio.

- B. The **Task Library** contains tasks that can be added to the process. The **Task Library** is divided into *drawers*. For information about tasks, see "Tasks Overview" on page 69.

  Click to collapse or expand the **Task Library**.

- C. **View Options**. **Graphical View** displays a graphical representation of the selected process. See “Graphical View” on page 27. **XML View** displays the XML code that defines the process. See “XML View” on page 27.

Templates and Instances

When you design a process in Process Studio, you are creating a process template that can be used to create an *instance*. An instance is a copy of a template that can be executed. Instances can be created in Process Studio for testing purposes, or (after a process has been published) executed on the APA Server. You can create more than one instance of a process.

The **Instances** tab displays instances that have been created for testing in Process Studio. Instances are grouped under top level nodes indicating the template from which they were created. ID numbers are included in instance names so that you can distinguish between multiple instances of the same process. The **Instances** tab is not displayed until an instance is created.

Graphical View

Graphical View provides a graphical representation of the process. It shows the Tasks that have been added to the process and how they are connected. Graphical View is the default view in Process Studio. Click the **Graphical View** tab to show the Graphical View.

XML View

XML View shows the XML that defines the current process. Click the **XML View** tab to see the XML that is saved in the Process definition (`.XPD`) file.

Process Studio includes a search feature that is used to search the XML markup.

To search the XML

1. On the **Edit** menu, click **Find in XML**. Alternatively, press CTRL + F.

   The **Find** dialog box opens.

2. In the **Search String** box, type the text you want to find.
3. Configure the search options:
   - Select the **Wrap Search** check box for the search to continue from the beginning when the end of the document is reached. The status bar indicates when the search has wrapped. Otherwise, you are notified that the end of the document has been reached.
   - Select the **Case Sensitive** check box to make the search case sensitive (distinguish between uppercase and lowercase text).

- Select the **Use Regular Expression** check box if you want to provide a regular expression instead of a literal search string. Further explanation of this syntax is available here: http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html

4. Click **Find Next**. If the Search String is found in the XML, the text is highlighted in the XML View.
5. Click **Find Next** repeatedly to continue searching, or click **Close** to cancel the search.

Tooltips

Tooltips are displayed when you hover the mouse pointer over a task in a template or instance. The Design, Comments, Duration, and Class properties are displayed in the tooltip.

Customize Process Studio

The Process Studio interface can be customized to make designing processes faster.

Customize the Default Scripting Language

Process Studio includes a Script Editor that enables you to write custom script for use in your processes. For information about using the Script Editor, see “Create Custom Scripts” on page 37. The default scripting language is the language that the Script Editor uses by default for new scripts.

To set the default scripting language

1. On the **Tools** menu, click **Options**.

   The **Options** dialog box opens.
2. In the **Default Scripting Language** list, click the language you want to use.
3. Click **OK**.

Customize the Task Library

You can add new drawers to the Task Library and define what tasks appear in each drawer.

To customize the Task Library

1. On the **Tools** menu, click **Options**.

The **Options** dialog box opens.

2. Click the **Gallery** tab.

   The drawers in the **Task Library** are displayed in the **Drawers** list.

   All of the Tasks that are available in Process Studio are displayed in **Tasks** and **Other** lists.

3. Customize the Task Library:
   - To add a task to a drawer, drag it from the **Tasks** list onto one of the folders.
   - To create a new drawer, click **New Drawer**.
   - To delete an existing drawer, click the drawer you want like to delete and click **Delete**.
   - To rename a drawer, click the drawer to select it. Wait a couple of seconds and click it again. Type the new name and press ENTER. The drawer is renamed. Tasks cannot be renamed.
   - Click **Restore Defaults** to return to the default settings for the entire **Options** dialog box. The settings on the **General** tab also revert to default.
4. Click **OK**.

Customize the Toolbar

You can choose whether to show or hide the toolbar, and adjust the size of buttons on the toolbar.

To show or hide the Toolbar

- On the **View** menu, click **Show Toolbar**.

To change the size of toolbar buttons

1. On the **Tools** menu, click **Options**.

   The **Options** dialog box opens.

2. Click either **Small** or **Large** icon size.
3. Click **OK**.

Process Studio Commands

This section describes the commands that are available in Process Studio.

- Open a New Window
- Zoom Controls
- Undo or Redo an Operation
- Show or Hide Grid Lines
- Compact Tasks
- View Messages
- Exit Process Studio

Open a New Window

To open an additional Process Studio window, use the **New Window** command. Creating a new window enables you to view different parts of the same process side-by-side. For example you can view the process template and a flow simulation at the same time.

The new window includes the same process templates and instances as the original window. Any changes you make in one window are also applied to other windows.

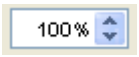
To open an additional Process Studio window

- On the **View** menu, click **New Window**.

  An additional Process Studio window opens.

Zoom Controls

To magnify the process displayed in the Graphical View, use the zoom controls.

You can type or select a zoom percentage in the **Custom zoom** 

box on the toolbar, or use the following procedure.

To zoom in or out

1. Click **View** > **Zoom**, and click one of the zoom options:
   - **Zoom In**. The process tasks appear larger.
   - **Zoom Out**. The process tasks appear smaller.
   - **Fit to Window**. The process is magnified so that all the Tasks fit in the Graphical View without scrolling
   - **100%**. The process is displayed at actual size.
   - **Custom**. You can define a custom zoom level.

The custom zoom dialog box opens.

i. Type a zoom percentage in the box (between 25 and 200%).

ii. Click **OK**.

Undo or Redo an Operation

Process Studio remembers the changes made to a process so that you can reverse your actions. Use the **Undo** command to undo the last operation. The Undo stack tracks the last 100 operations. Not all operations can be reversed.

To undo the last operation

- On the **Edit** menu, click **Undo**.
- Press CTRL + Z.

To redo the last undone operation

- On the **Edit** menu, click **Redo**.
- Press CTRL + Y.

Show or Hide Grid Lines

To show or hide the grid lines in the process template and instance backgrounds, use the **Grid** command.

To show or hide the grid

- On the **View** menu, click **Grid**.

Compact Tasks

To layout the Tasks in the current Process Template so that they use the minimum amount of space, use the **Pack** command.

To layout tasks using the minimum space

- On the **Edit** menu, click **Pack**.

View Messages

The Console is used to view messages regarding startup problems or unexpected application errors.

To open the console

- On the **View** menu, click **Console**.

Exit Process Studio

To exit Process Studio, use the following procedure.

To exit Process Studio

1. On the **File** menu, click **Exit**.
2. If there is a single unsaved process templates, a dialog box opens asking if you want to save your work.
   - Click **Yes** to save the process template and exit Process Studio.
   - Click **No** to exit Process Studio without saving.
   - Click **Cancel** to keep Process Studio open.

   If there are multiple unsaved process templates a dialog box opens that lists the unsaved process templates. You can choose which templates you want to save.
   - Click **Select All** to save all the unsaved process templates or **Deselect All** to discard them. Click **OK** to confirm your selection.
   - Alternatively, select or clear the check boxes in the **Save** column to save or discard specific process templates. Click **OK** to confirm your selection.
   - Click **Cancel** to keep the application open and not save the selected process templates.
3. If you have chosen to save one or more processes that have not been saved before, the **Save** dialog box opens. See “Save As” on page 48.

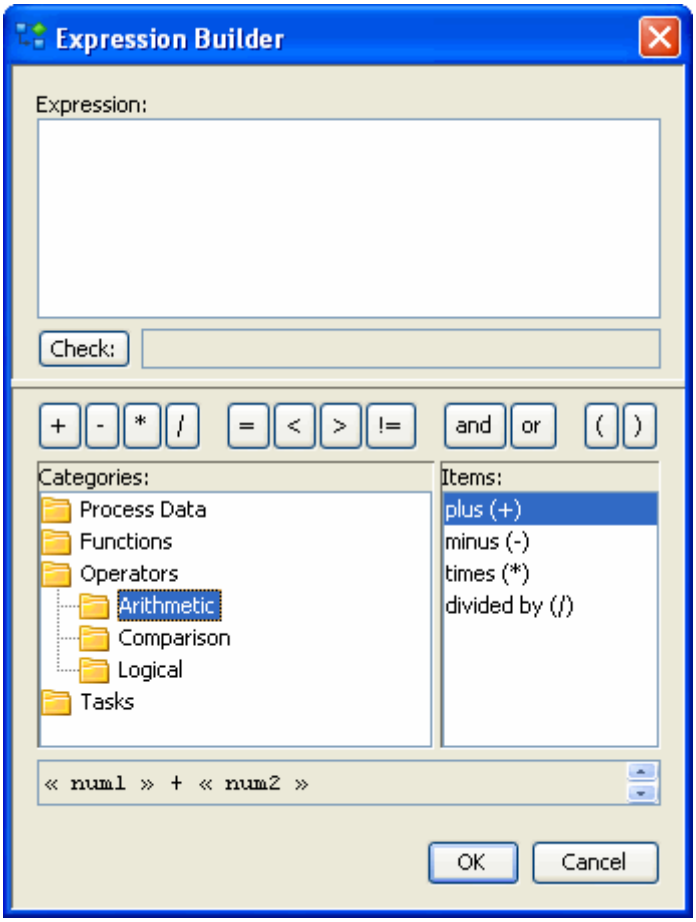
Build Expressions

The Expression Builder is a powerful tool that is available in APA Form Designer and APA Process Studio.

In Process Studio, the Expression Builder can be used to create expressions that control the behavior of Branch and Loop tasks. The Expression Builder is opened from the task **Properties** dialog box (see “Configure a Branch Task” on page 100 and “Configure a Loop Task” on page 102).

In many cases the Expression Builder can save you from writing custom scripts. To use the Expression Builder, you must understand the APA Expression Language. See “APA Expression Language” on page 171.

Figure 1 The Expression Builder User Interface



The Expression Builder user interface consists of:

- **Expression box**. The Expression box displays the expression. Type in the box to create or modify an expression.
- **Check button**. Click **Check** to test the expression for validity. Any syntax errors are reported.

  The expression is also checked when you click **OK**.
- **Operator buttons**. Click a button to insert the operator into the expression at the current insertion point. Clicking a button is the same as typing the operator directly into the **Expression** box.
- **Categories list**. Displays a list of folders. Each folder contains elements that can be inserted in your expressions.

  Click a folder to show or hide its subfolders and display a list of items in the **Items** list.

- **Process Data**. Contains subfolders for Process Attributes, Fields, and Process Properties that can be used in the expression.

  Refer to the *APA Programming Guide* for more information about Process Data and Process Properties.

- **Functions**. Contains subfolders for the functions defined by the Expression Language. See "APA Expression Language" on page 171.
- **Operators**. Contains subfolders for the operators defined by the Expression Language. See "APA Expression Language" on page 171.
- **Tasks**. Contains subfolders for each task on the current process template. You can insert task attributes and properties into your expressions by double-clicking the relevant item in the **Items** list. For more information about defining task properties and attributes, see "Configure Tasks" on page 91. For more information about Task Properties refer to the *APA Programming Guide*.

- **Items list**. Displays the expression elements for the folder selected in the **Categories** list.

  Double-click an element to insert it into the expression at the current insertion point. The effect is the same as if you had typed the element into the expression box.

- **Description area**. The Description area is located below the **Categories** and **Items** lists. Descriptive text is displayed for the item selected in the **Items** list.

Create Dynamic Values Using Substitution Syntax

Substitution syntax enables the creation of dynamic values. When Process Studio reads Substitution Syntax, it replaces it with a real value from the process. Substitution Syntax can be typed in all of the following locations:

- E-mail task properties.
- Manual task properties: **Web Link** and **Instructions**.
- Message task properties: **Web Link** and **Text**.
- Command task properties: the command itself, and every element of the argument list.
- Web Service task properties: in the request string.

NOTE If a field with multiple values (that is, from a multi-select list) is referenced in a substitution, only the first value is used in the substitution

Substitution Syntax Example

A process is created for creating and approving purchase orders. The process originator completes a purchase order form that is then assigned to a manager for approval. When the purchase order is approved or rejected, an E-mail task is used to send an e-mail to the process originator.

The following substitution syntax might be used in the e-mail message:

`Purchase Order # $Field(Order_Number) has been rejected. Contact your supervisor for additional information.`

When the e-mail is sent, the text **`$Field(Order_Number)`** is replaced with the order number used on the purchase order form.

Substitution Syntax Definitions

Sequence	`$Attr(attrName)`
Description	Retrieves the value of the `attrName` process attribute as a string. Returns an empty string if the attribute does not exist or has an empty value.
Examples	`$Attr(purpose)` – gets the value of the *purpose* attribute

Sequence	`$TaskAttr(scopedId, attrName)`
Description	Retrieves the value of the `attrName` task attribute as a string. The `scopedId` parameter must refer to a scoped task identifier on the current process template. Returns an empty string if the scoped identifier is invalid, if the attribute does not exist, or if the attribute has an empty value.
Examples	`$TaskAttr(123, comments)` –returns the value of the *comments* attribute on the task with a scoped ID of 123.

Sequence `$Prop(propertyName)`

Description Retrieves the value of the specified process property as a string. The `propertyName` parameter is any string ‘xyz’ where the Process class (as defined in the new BPM API) has a corresponding `getXyz()` method. Returns an empty string if the property name is invalid or the property has an empty value.

Examples `$Prop(originator)` – gets the originator of the process

`$Prop(startDate)` – gets the start date of the process

Sequence `$TaskProp(scopedId, propertyName)`

Description Retrieves the value of the specified task property as a string. The `scopedId` parameter must refer to a scoped task identifier on the current process template. The `propertyName` parameter may be any string ’xyz’ where the Task class (as defined in the new BPM API) has a corresponding `getXyz()` method. Returns an empty string if the scoped identifier is invalid or if the property name is invalid or the property has an empty value.

Examples `$TaskProp(123, performer)` – returns the performer of the task with a scoped ID of 123

`$TaskProp(456, startDate)` – returns the start date of the task with a scoped ID of 456

Sequence `$Field(fieldName)`

Description Retrieves the value of the specified process field as a string. Returns an empty string if the field does not exist or has an empty value.

Examples `$Field(firstname)` – gets the value of the *firstname* field

Sequence `$TaskField(scopedId, fieldName)`

Description Retrieves the value of the specified task field as a string. The `scopedId` parameter must refer to a scoped task identifier on the current process template. Returns an empty string if the scoped identifier is invalid or if the field does not exist or has an empty value.

Examples `$TaskField(123, lastname)` – returns the value of the *lastname* field on a task with a scoped ID of 123

Additional Information

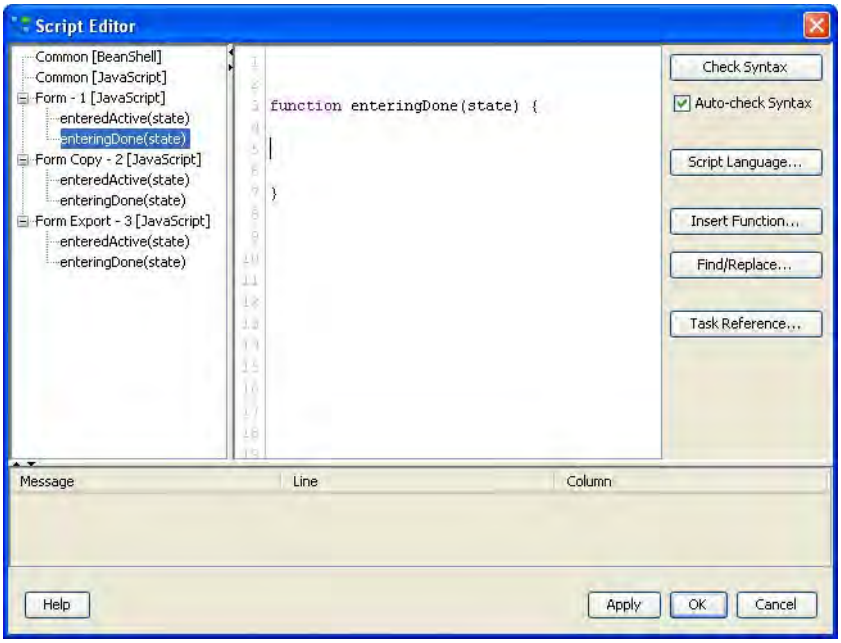
- A substitution sequence can be escaped by using the backslash '\' character.
- Parameters used in the substitution syntax are not quoted.
- When a literal '$' character precedes a pattern ALPHA (where ALPHA can be any one or more alphabetic characters, the '$' character must be escaped by a preceding '\' character. For example, the string "Price is listed in \ $USD(current)" would be treated as the string "Price is listed in $USD(current)".
- When a '$' character appears outside of a substitution sequence, no escaping is necessary. For example, the string "The price is $4" would be treated as a literal string and would not be substituted.
- The '\' escape character can itself be escaped by adding another '\' character. For example, the string "Name is \\$attr(name)" would be substituted with the string "Name is \x" where 'x' is the value of the 'name' attribute.
- When an attribute or property is referenced in substitution syntax, the value of the attribute or property is always converted to a string for the substitution.

Create Custom Scripts

Process Studio includes a Script Editor that enables you to write custom script for your processes. You can use JavaScript and BeanShell solutions. Refer to the *APA Programming Guide* for information about writing custom script.

NOTE The Script Editor is designed for users who have a basic understanding of JavaScript, BeanShell, and programming concepts.

The following image shows the Process Studio Script Editor:



To create a script

1. Open the Script Editor:
   - On the **Tools** menu, click **Script Editor**.
   - Right click a task in the Graphical view and click **Task Properties**. Click the **General** tab. Select the **Enable script** check box and click **Edit**.

   The **Script Editor** dialog box opens.
2. Write your script.
3. To save your script, click **Apply**. To save and exit the Script Editor, click **OK**. To close the Script Editor without saving the script, click **Cancel**.

Related Topics

- "Check a Script for Syntax Errors" on page 39
- "Set the Script Language in the Script Editor" on page 39
- "Find and Replace Text in the Script Editor" on page 39
- "Insert a Function into a Script" on page 40
- "Insert a Task Reference into a Script" on page 40

Check a Script for Syntax Errors

Any script errors cause a task to go to the Abort state. It is best to use an Exception task to handle errors in script.

To check a script for syntax errors, use the following procedure.

To check a script for syntax errors

- Click **Check Syntax** to check the script for errors.
- Select the **Auto-check Syntax** check box for the Script Editor to check the script for syntax for errors as you type.

  Script errors are displayed in the list at the bottom of the Script editor.

Set the Script Language in the Script Editor

To set the script language in the Script Editor, use the following procedure.

To set the script language in the Script Editor

1. Click **Script Language** to define the language for the current script.

   The **Script Language** dialog box opens.
2. In the **Script Language for This Script** list, click one of the languages.
3. If the selected language is not the default language, you can select the **Set Selected Language as Default for New Tasks and Processes** check box to make it the default language
4. Click **OK**.

Find and Replace Text in the Script Editor

To find and replace text in the Script Editor, use the Find/Replace tool.

To find and replace text in the Script Editor

1. Click **Find/Replace**.

   The **Find/Replace** dialog box opens.
2. In the **Search String** box, type the text you want to search for.
3. If you would like to replace the text, type the text you want to use as a replacement in the **Replace With** box.
4. Configure the search options:
   - Select the **Wrap Search** check box to continue the search from the beginning when the end of the script is reached.

- Select the **Case Sensitive** check box to make the search case sensitive (distinguish between uppercase and lowercase text).
- Select the **Use Regular Expression** check box if you want to provide a regular expression instead of a literal search string. Further explanation of this syntax is available here: http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html

5. Click **Find Next**. If the text in the **Search String** box is found in the script, it is highlighted in the Script Editor.
6. To ignore the result and search again, click **Find Next**.

   To replace the text and search again, click **Replace/Find Next**.

   To replace every occurrence of the **Search String**, click **Replace All**.

   To close the Find/Replace dialog box, click **Close**.

Insert a Function into a Script

To insert a function into a script, use the following procedure.

To insert a function into a script

1. Click **Insert Function**.

   The **Insert Function** dialog box opens.
2. In the list, click the function you want to insert.
3. Click **OK**.

TIP You can also insert a function by double-clicking the function name in the Script Editor navigation tree. If the function already exists, the cursor moves to the location of the existing function.

Insert a Task Reference into a Script

To insert a task reference into a script, use the following procedure.

To insert a task reference into a script

1. In the Script Editor, place the cursor in the script where you would like to place a reference to a task.
2. Click **Task Reference**.

   The **Select Task** dialog box opens.

3. Type an appropriate Attribute Name in the **Attribute Name** box. The default Attribute Name is the name of the task.
4. **Click OK**.

   An attribute is created on the task or process that contains the Scoped ID of the selected task. A string is inserted at the current cursor position within the script that uses the attribute value to retrieve a task reference.

PART 2 Create and Deploy Processes

This section provides information about creating, testing and publishing Processes.

- Create and Save a Process
- Configure a Process
- Tasks Overview
- Add and Connect Tasks
- Configure Tasks
- Simulate a Process
- Connect to an APA Server
- Publish a Process

CHAPTER 3

Create and Save a Process

A process is set of connected tasks that are designed to achieve a specific goal. This section describes how to:

- Create a Process
- Create a New Process
- Open a Process
- Retrieve a Process from the APA Server
- Save a Process
- Save a Process as an Image
- Print a Process
- Close a Process

Create a Process

Process Studio enables process designers to quickly create, test, and publish processes. A typical workflow for creating and publishing a process includes the following steps:

1. Create a new process in Process Studio. See “Create a New Process” on page 46.
2. Add tasks to the process. See “Add a Task” on page 81.

3. Connect the tasks in a logical sequence. See “Connect Tasks” on page 83.
4. Configure the tasks. See “Configure Tasks” on page 91.
5. Configure the process. See “Configure Process Properties” on page 51.
6. Test the process. Process Studio includes features that can be used to test a process before it is deployed to the APA server. See “Load Simulation” on page 127.
7. Connect to the APA Server. See “Connect to the APA Server” on page 131.
8. Publish the process. Publishing includes saving the process to the APA server, and defining permissions for the process. See “Publish a Process” on page 137.

Process Studio Tutorial

For a detailed guide that describes how to create a process using Process Studio, see “Process Studio Tutorial” on page 185.

Create a New Process

When Process Studio starts, a new blank process is created. To create another new process, use one of the following commands.

To start a new, blank process template

- On the **File** menu, click **New**.
- Click **New** .
- Press CTRL + N.

Related Topics

- “Configure Process Properties” on page 51
- “Add a Task” on page 81
- “Open a Process” on page 46
- “Save a Process” on page 48

Open a Process

To open an existing process template, use the **Open** command.

To open an existing process template

1. To open an existing process template:
   - On the **File** menu, click **Open**.
   - Click **Open** .
   - Press CTRL + O.

   The **Open** dialog box opens.
2. Navigate to, and click, the process template that you want to open.
3. Click **Open**.

Related Topics

- “Create a New Process” on page 46
- “Retrieve a Process from the APA Server” on page 47
- “Save a Process” on page 48

Retrieve a Process from the APA Server

A process template can be retrieved from the APA server and opened in Process Studio. The folders and processes that you can retrieve depend on the permissions configured by your APA Administrator.

To retrieve a process template from the APA Server

1. Connect to the APA Server (see “Connect to the APA Server” on page 131).
2. On the **File** menu, click **Retrieve Process**.

   The **Retrieve Process** dialog box opens.
3. Select a process template. See “Search the APA Server” on page 132.

Related Topics

- “Create a New Process” on page 46
- “Open a Process” on page 46
- “Save a Process” on page 48

Save a Process

To save a process template to an XML Process Definition (.XPD) file, use the **Save** command.

To save a process template

1. There are several ways to save a process template:
   - On the **File** menu, click **Save**.
   - Click **Save** .
   - Press CTRL + S.
2. If the file has not been saved before, the **Save** dialog box opens. See "Save As" on page 48.

Related Topics

- "Create a New Process" on page 46
- "Open a Process" on page 46
- "Save a Process as an Image" on page 49

Save As

To save the process template to a new XML Process Definition (.XPD) file, use the **Save As** command.

To save a process template to a new file

1. On the **File** menu, click **Save As**.

   The **Save** dialog box opens.
2. Navigate to the location where you want to save the file.
3. Type a **File Name**.

   If the **Use File Name as Process Title** check box is selected, the File Name is used as the Process Title.
4. Click **Save**.

   The process template is saved to the selected location.

Related Topics

- "Create a New Process" on page 46
- "Open a Process" on page 46

- “Save a Process as an Image” on page 49

Save a Process as an Image

To create an image file containing a visual representation of the process template, use the **Save As Image** command.

To save a process template as an image

1. On the **File** menu, click **Save As Image**.

   The **Save** dialog box opens.
2. Browse to the location where you want to save the file.
3. In the **File Name** box, type a name for the image.
4. Click **Save**.

   The image file is saved to the selected location.

Related Topics

- “Save a Process” on page 48
- “Print a Process” on page 49

Print a Process

To print the process template, use the **Print** command.

To print a process template

1. Click the print command:
   - On the **File** menu, click **Print**.
   - Click **Print** .
   - Press CTRL + P.

   The **Print** dialog box opens.
2. If required, set the print range, number of copies, and change the printer properties.
3. Click **Print**.

Related Topics

- “Save a Process as an Image” on page 49

Close a Process

To close the current process template, use the **Close** command.

To close a process template

1. On the **File** menu, click **Close**.
2. If the process has not been saved, a dialog box opens.

   Click **Yes** to save the changes and close the process template.

   Click **No** to close without saving the changes.

   Click **Cancel** to keep the process template open.

Related Topics

- “Create a New Process” on page 46
- “Open a Process” on page 46
- “Save a Process” on page 48

CHAPTER 4

Configure a Process

A process is a connected set of process tasks that are designed to achieve a specific goal. This section describes how to configure a process.

- Configure Process Properties
- Form Packets
- Form Packet Rules

Configure Process Properties

Process Properties describe a process and define its behavior.

Process Properties can be modified for either a template or an instance. If you change the properties of an instance, the changes do not affect the process template.

To set up process template properties

1. Open the process properties dialog box:
   - On the **Edit** menu, click **Process Properties**.
   - Press ALT + SHIFT + ENTER.
   - Right-click a blank area of the **Graphical View** and click **Process Properties**.
2. Modify the properties as required:

- To describe the process (title, author, description and keywords) and configure script options, use the **General** tab. See “General Process Properties” on page 52.
- To assign or modify process attributes, use the **Attributes** tab. See “Configure Attributes” on page 53.
- To view or create process fields, use the **Fields** tab. See “View and Configure Process Fields” on page 54.
- If you want to create and configure Form Packets for use in your process, use the **Packets** tab. See “Form Packets” on page 57.
- To configure the Initial form properties, use the **Initial Form** tab. See “Configure Initial Form Properties” on page 56.

3. Click **OK**.

Related Topics

- “Create a New Process” on page 46
- “Add a Task” on page 81

General Process Properties

The **General** Tab of the process properties dialog box includes the following properties:

Title	The process title. This is the default used when you publish the process to the APA Server.
Author	The name of the process author.
Description	A description of the process.

Keywords	Keywords that describe for the process (separated by commas).
Date Estimation Enabled	If this check box is selected the APA server calculates data about projected durations for unexecuted tasks. Enabling this option could potentially impact performance.
Script properties	To enable script for this process, select the **Enable Script** check box. To edit the script in the Script Editor, click **Edit**. See "Create Custom Scripts" on page 37. To enable script in Flow Simulation Mode, select the **Enable Script in Simulation** check box. (This check box is only available if **Enable Script** is selected).

NOTE It is important to type a **title**, **author**, **description**, and **keywords** for the process so that it can be found when searching the APA server.

Configure Attributes

Attributes can be assigned to the process using the **Attributes** tab of the process properties dialog box.

Add an attribute

To add an attribute to the process, use the following procedure.

To add an attribute

1. Click **New**.
2. In the **Name** column, type a name for the attribute.
3. In the **Value** column, type a value for the attribute.
4. In the **Type** column, click a data type (**Text**, **Number**, or **Scoped ID**).

NOTE A Scoped ID is a task identifier that is unique within a process.

Delete an attribute

To delete an attribute, use the following procedure.

To delete an attribute

1. Click the attribute in the list.
2. Click **Delete**.

View and Configure Process Fields

The **Fields** Tab of the process properties dialog box displays all forms, packet namespaces and custom fields in the process.

You can add and delete custom fields in the process definition. You can delete any field from the list, but if the field still exists on a form, it is automatically recreated the next time the field list is refreshed. Custom fields do not appear on forms and are not recreated.

Add a Custom Field to the Process Definition

To add a custom field to the process definition, use the following procedure.

To add a custom field to the process definition

1. Click **New**. A field is added to the bottom of the list.
2. In the **Name** column, type the field name.
3. In the **Value** column, type the field value.
4. In the **Type** column, click a data type in the list.
5. To make a field Sticky (read-only), select the check box in the **Sticky** column.

Delete a Field from the Process Definition

To delete a field from the process definition, use the following procedure.

To delete a field from the process definition

1. Click the field in the list.
2. Click **Delete**.

Data Types for Process Fields

Process fields can use one of the following data types:

Data Type	Storage Format
Number	The field value is stored as a number with '.' as a decimal symbol. All currency symbols and other separators are removed.
Text	The field value is not formatted.
Date	The field value is stored as 'CCYY-MM-DD'.
Time	The field value is stored as 'HH:MM:SS' (24-hour).

When process field values are used in forms, the values are displayed using form-specific formatting. When a form task is completed, form field values are reformatted to store process field values in the format described. When process field values are used only in conjunction with APA Servers, the user designing the process does not need to be concerned about the storage format of process fields. However, the user does need to be aware of the internal formatting in the following cases:

- Getting process field values from script always returns the value in the internal format.
- To set the value for a process field in script, the value must conform to the internal format based on the data type of the process field.
- If a process is initiated with initial values for process fields (by sending extra parameters on the URL or using SOAP or Virtual Submit), the initial values must conform to the internal format.
- If specifying values for process fields in the `setprocessTaskDone` SOAP call, the values must conform to the internal format.
- The internal format is used when substituting process field values in for task parameters (see "Create Dynamic Values Using Substitution Syntax" on page 34).

The APA Server automatically determines the data type for any process field that corresponds to a form field of a form included in the process.

- If the field exists on only one form included in the process, the data type for that form field is used as the data type for the process field.
- If the field exists on multiple forms included in the process and the field has the same data type on each form, that data type is used for the process field.
- If the field exists on multiple forms but the field does not have the same data type on each form, the data type for the process field is set to String.

Process Studio contacts the APA Server to determine process field data types at the following times:

- On listing process fields in the process field configuration screen.
- On opening the **Expression Builder**.
- On publishing the process.

You can also manually set the data type for a process field.

After the data type for a particular field has been set either manually or automatically, no more automatic checking occurs for that field. If any changes are made that affect the data type, the process designer must manually change the data type. Some possible changes that can affect the data type include:

- Introducing a new form into a process definition.
- Republishing a form that is part of a process definition.

Configure Initial Form Properties

The **Initial form** tab of the process properties dialog box configures the Initial form properties. The first task in many processes is a Form Task, which is assigned to a user to enter data.

To configure Initial form properties

1. Select the **Use Initial Form** check box.
2. Click one of the instantiation options:
   - **Instantiate Process Before Displaying Form**. The server starts the process immediately before displaying the initial form.
   - **Instantiate Process After Submitting Form**. If you select this option, the following occurs:
     - The APA Server launches the form specified in the selected **Initial Task** list but does not start a new process.
     - As the process has not been created, there are no process script entry points available to perform setup for the initial form task. The form and initial task script entry points are executed as with any other form task.
     - The process Progress Map and History screen in the APA Portal and APA Web Desktop are not available until after the process starts.
     - The form may be saved for later and used to launch a process at a later time after it is submitted.
3. In the **Initial Task** list, click an initial task. The Initial task is the task in the process that should be executed first.

NOTE To define a Form Packet as an Initial Task you must click **Instantiate Process Before Displaying form**.

If the process is instantiated before the form is displayed, any Form task in the process that is assigned to *Originator* can be used. If the process is instantiated after the form is displayed, the task and all of its parent tasks must not have any predecessor tasks.

Published processes that have an Initial Task selected display form icons (HTML, PDF) in the Portal and Web Desktop, rather than the process icon normally displayed when processes are published to the APA Server.

Form Packets

A Form Packet is a collection of forms that are bound together for processing.

Use the **Packets** tab of the process properties dialog box to create and configure Form Packets.

NOTE This option is not available in all versions of Process Studio.

Related Topics

- “Form Packet Task” on page 77.

Create and Configure a New Form Packet

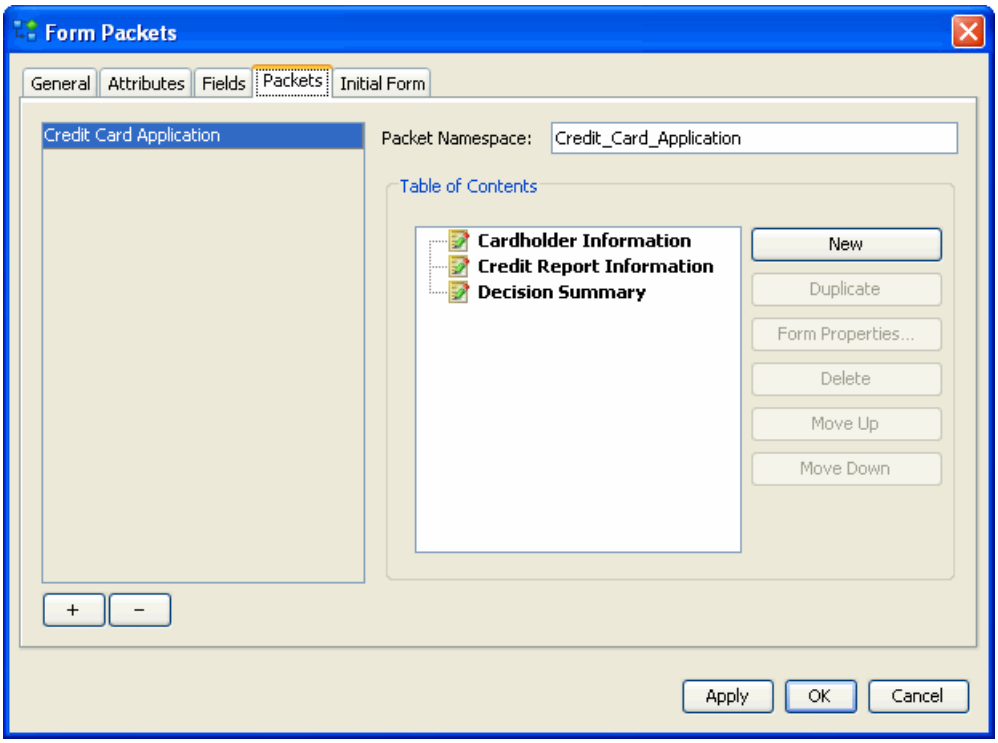
To create and configure a new form packet, use the following procedure.

To create and configure a new Form Packet

1. On the **Edit** menu, click **Process Properties**.

   The process properties dialog box opens.

2. Click the **Packets** tab.



3. Click + .

   The **Enter Packet Name** dialog box opens.

4. Type a name for the packet and click **OK**.
5. Use the **Table of Contents** area to configure the structure of the packet.
   - To add a new node
     i. Click **New**.
     ii. Type a name for the node and press ENTER.
   - To add a new child node, add a new node and then drag the node onto an existing node. The new node becomes a child of the existing node.
   - To duplicate a node, click **Duplicate**. Any form properties, child nodes and child properties are also duplicated.
   - To delete a node, select the node and click **Delete**.
   - To move a node in the structure, click **Move Up** or **Move Down**.
6. To associate a form with a node, click the node and click **Form Properties**.

   A dialog box opens.

   a. In the **Default Namespace** box, select the default namespace.

- **Path**. Uses the path within the Table of Contents tree starting with the Packet Namespace and including the selected node.
- **Parent**. Uses the path within the Table of Contents tree starting with the Packet Namespace and ending with the parent of the selected node.
- **Root**. Uses the predefined Packet Namespace.

b. Click **Search** to select a form on the APA server. See “Search the APA Server” on page 132.

c. If required, edit the **Namespace** and **Storage Name** for the form fields. Like any other process attribute, form packet fields can be referenced outside of form packet tasks (such as from a regular form task) by specifying the full namespace. Form packet namespaces have the structure `Packet$Folder$Subfolder$Form`.

d. Select or clear the **Represent Static Table Cells as Top Level Fields** check box. For more information, see “Represent Static Table Cells as Top Level Fields” on page 121.

e. Click **OK**.

Delete a Form Packet

To delete a form packet, use the following procedure.

To delete a Form Packet

1. On the **Edit** menu, click **Process Properties**.

   The process properties dialog box opens.
2. Click the **Packets** tab.
3. Click the name of the Form packet in the list.
4. Click .

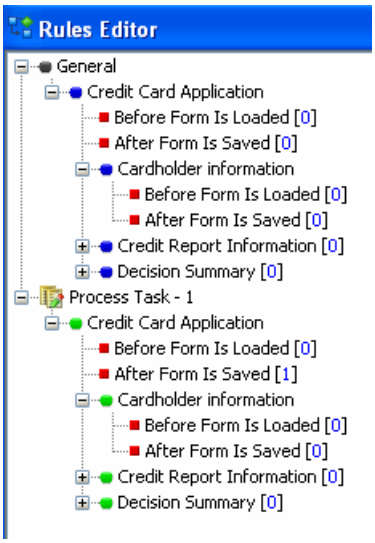
Form Packet Rules

Rules for Packets and Form Packet tasks are created using the Rules Editor. Rules are applied based on *Conditions*. Conditions can be defined using an Expression or Script.

Consequences are the results of a rule being applied. The Rules Editor includes the following pre-defined consequences:

Stop Rule Execution	Stops executing rules for this event instance. For example, if a form were saved, when this consequence was encountered, rule execution would stop. If the form were saved again, rules would trigger normally.
Restart Rule Execution	Restarts rules for this event. For example, if there were five rules and this were the consequence of the third, execution would begin again at the first rule. The presumption is that some action has been taken to prevent this rule consequence from triggering on the second run.
Script	Executes the provided script.
Set Node State	Sets the display state of a node in the packet at this task.
Set Message in TOC	Displays a message in the table of contents above the selected node.
Add Form	Adds a form to the packet for all tasks.
Remove Nodes	Removes one or more nodes from the packet for all tasks.
Duplicate Node	Duplicates a node on the packet for all tasks.
Chain to Next Node	Chaining specifies which form should be opened next. “Chain to Next Node” specifies that the next node should be opened (ignoring nodes that are not associated with forms). Chaining must be enabled by a separate consequence if not already enabled.
Chain to Previous Node	Chaining specifies which form should be opened next. “Chain to Previous Node” specifies that the previous node should be opened (ignoring nodes that are not associated with forms). Chaining must be enabled by a separate consequence if not already enabled.
Chain to Specific Node	Chaining specifies which form should be opened next. “Chain to Specific Node” allows you to choose the node that is opened next. Chaining must be enabled by a separate consequence if not already enabled.
Set Chaining	Enables or disables chaining.

The Rules Editor allows you to define rules for a form packet (using the **General** section), or for specific process tasks that reference the form packet:



Related Topics

- “Create a Rule” on page 61.

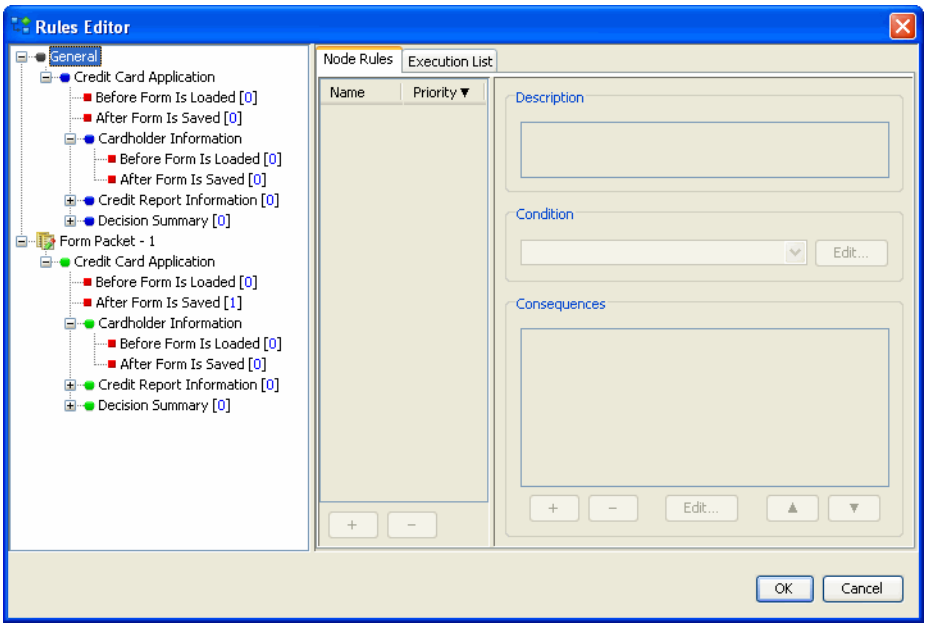
Create a Rule

To create a rule, use the following procedure.

To create a node Rule

1. On the **Tools** menu, click **Rule Editor**.

   The **Rules Editor** dialog box opens.



2. Click the **Node Rules** tab.
3. Click + underneath the **Node Rules** list.

   A new rule is added to the list.
4. In the **Name** column, type a name for the rule.
5. In the **Priority** column, set the priority.
6. In the **Description** box, type a description for the node.
7. Specify when the rule is applied:
   a. In the **Condition** list, click either **Expression** or **Script** to specify a condition.
   b. Click **Edit** to edit the expression or script.
8. In the **Consequences** area, specify the consequences of the rule being applied.
   - To add a new consequence, click +.

     The **Consequence** dialog box opens.

     i. In the **Consequence** list, click a consequence.

     ii. If required, change the **Display Name** of the consequence.

     iii. Click **OK**.
   - To remove a consequence, click −.
   - To configure a consequence, click the consequence and click **Edit**.

A dialog box opens for you to configure the consequence. See the following topics:

- “Configure the Set Node State Consequence” on page 63.
- “Configure the Set Message in TOC Consequence” on page 64.
- “Configure the Add Form Consequence” on page 65.
- “Configure the Remove Nodes Consequence” on page 67.
- “Configure the Chain to Specific Node Consequence” on page 68.
- If the consequence is a script, the script editor opens so that you can write custom script. For information about writing custom script, see “Create Custom Scripts” on page 37 and refer to the *APA Programming Guide*.

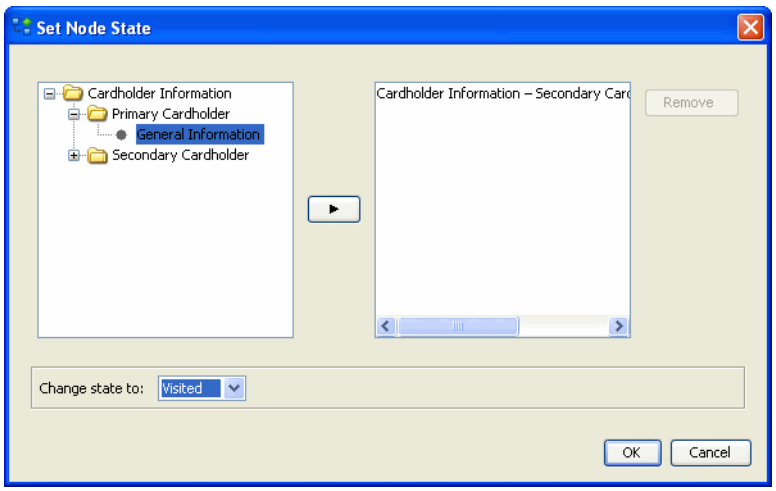
9. To view the list of rules that are applied, click the **Execution List** tab.

   The execution list shows the list of rules that are applied when the selected event occurs.

10. Click **OK**.

Configure the Set Node State Consequence

The “Set Node State” consequence changes the state of a node in a Form Packet.



To configure the Set Node State consequence

1. In the left pane, click the node that you want to change state, and click **>**. (You can select multiple nodes).
2. In the **Change state to** list, click one of the following options:

Visited	Node has been visited on this task. This is set automatically when an unvisited form is displayed.
Unvisited	Node has not been visited. This is the default state.
Inactive	The node is not active.
Error	An error exists on the node.
Hidden	Node is not displayed.

The state that you choose will apply to all of the nodes displayed in the right pane.

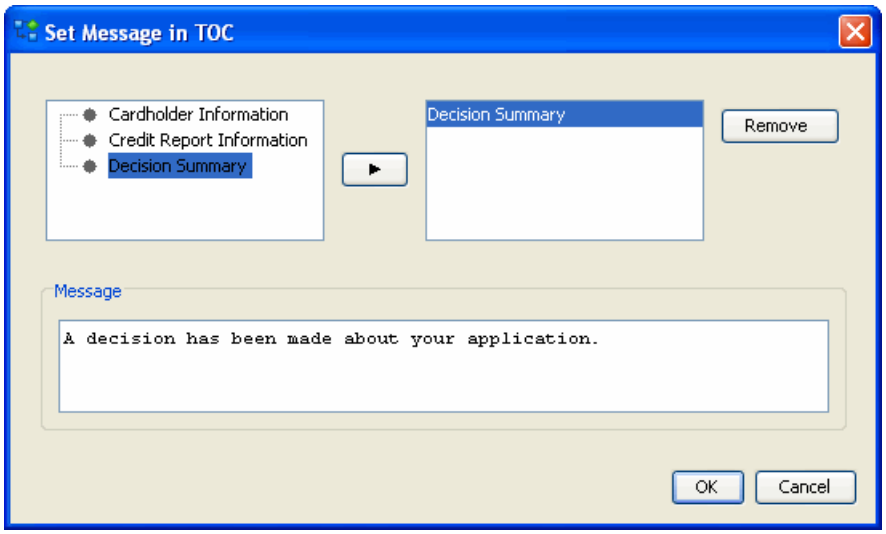
3. Click **OK**.

Related Topics

- “Create a Rule” on page 61.

Configure the Set Message in TOC Consequence

The “Set Message in TOC” consequence adds a message to the table of contents of a form packet. This message is displayed when the table of contents is viewed in Portal or Web Desktop.



To configure the message

1. In the left pane, select the nodes in the table of contents that should display the message and click **>**.

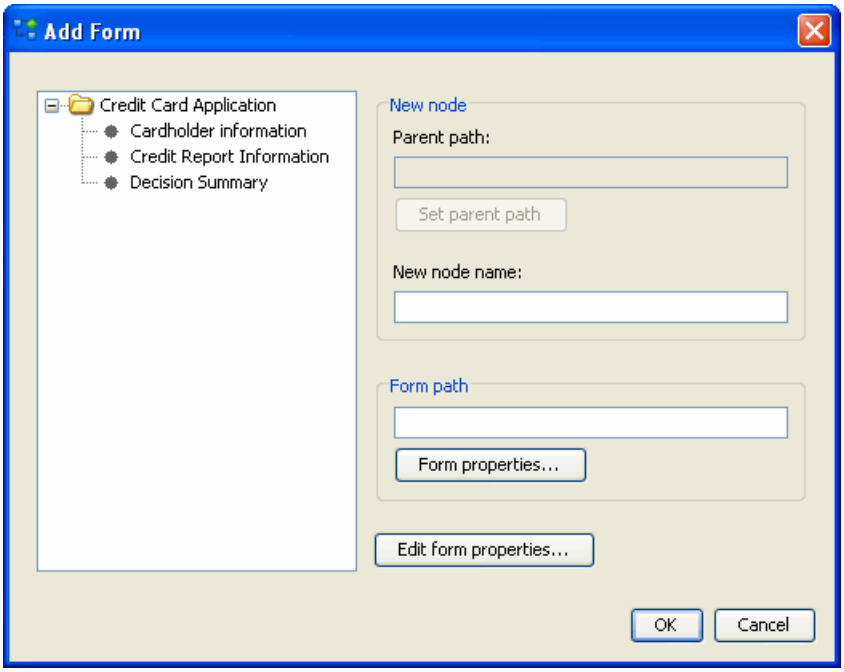
2. In the **Message** box, type the message that you want to display.
3. Click **OK**.

Related Topics

- “Create a Rule” on page 61.

Configure the Add Form Consequence

The “Add Form” consequence adds a form to the form packet.



To configure the Add Form consequence

1. Choose the parent path of the new node.
   a. In the list, click the node that should contain the new form.
   b. Click **Set Parent Path**.
2. In the **New node name** box, type a name for the new node.
3. In the **Form path** area, specify the form that should be added to the packet. You can either type the path of a form, or use the following steps:
   a. Click **Form Properties**.

      A dialog box opens.
   b. Click **Search** and select a form from the APA server.

c. Click **OK**.

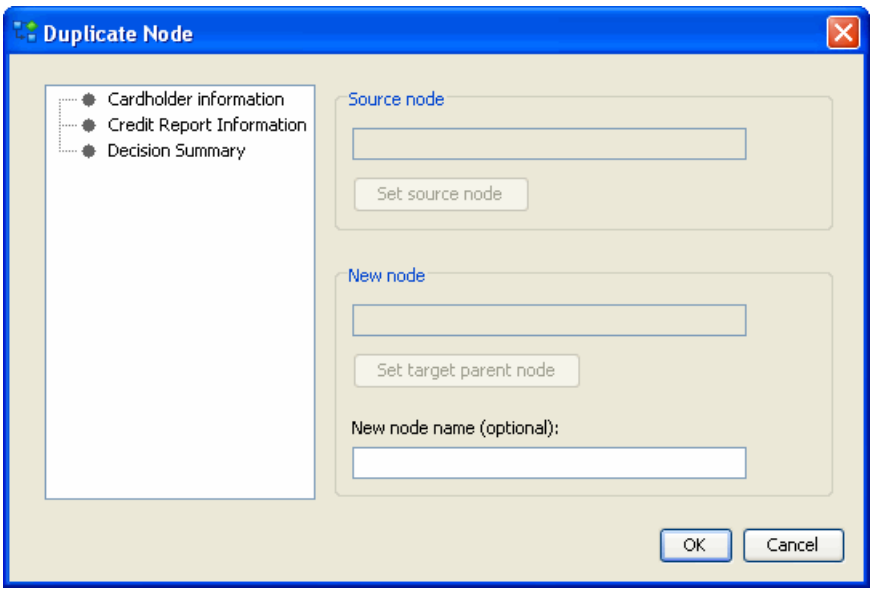
4. If you want to edit the properties of the form, click **Edit form properties**. For information about each property, see the following topics:
   - “Allow Form to be Edited” on page 117
   - “Initialize Form at this Task” on page 118
   - “Set Field Properties” on page 117
   - “Customize Submit Action List” on page 118
   - “Enable Form on Mobile devices” on page 119
   - “Automatically Display Next Form on Submit” on page 121
5. Click **OK**.

Related Topics

- “Create a Rule” on page 61.

Configure the Duplicate Node Consequence

The “Duplicate node” consequence duplicates the selected node in the form packet.



To configure the Duplicate Node consequence

1. In the list, click the node that you want to duplicate and click **Set source node**.
2. Specify the location of the new node.

- To place the duplicated node under another node, click that node in the list and click **Set target parent node**.
- To place the duplicated node in the root of the form packet, do not specify a target parent node.

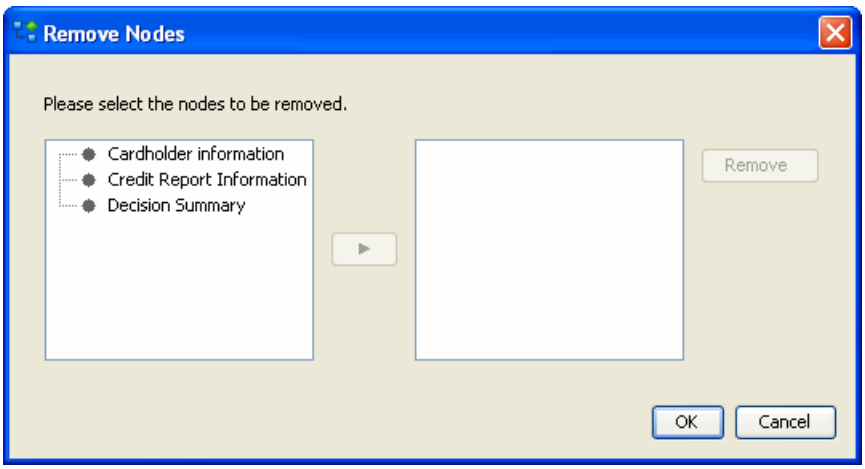
3. In the **New node name** box, type a name for the new node. The name is optional. If you do not specify a name, the name of the original node is used.
4. Click **OK**.

Related Topics

- “Create a Rule” on page 61.

Configure the Remove Nodes Consequence

The “Remove nodes” consequence removes one or more nodes from a form packet.



To configure the Remove Nodes consequence

1. In the list, click the nodes that you want to remove. To select multiple nodes, hold down CTRL and click each node.
2. Click **>**

   The nodes are added to the other list.
3. Click **OK**.

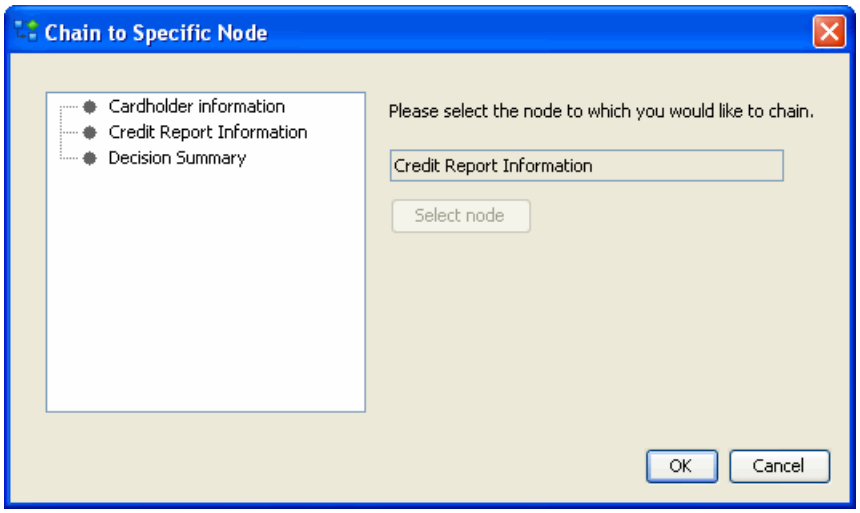
Related Topics

- “Create a Rule” on page 61.

Configure the Chain to Specific Node Consequence

The “Chain to Specific Node” consequence allows you to select the next form that should be displayed to the user. When forms are chained, and a user submits a form, the next form is displayed automatically.

Using rules to specify chaining allows you to bypass the table of contents for a form packet. The order of the forms can be changed, or nodes ignored, depending on the data that is entered on a form.



To configure the Chain to Specific Node consequence

1. In the list, click the node that you want to display next.
2. Click **Select node**.

   Click **OK**.

Related Topics

- “Create a Rule” on page 61.

CHAPTER 5

Tasks Overview

This section describes the tasks that are available in Process Studio.

- Tasks Overview
- Process Control Tasks
- Form Tasks
- Other Tasks

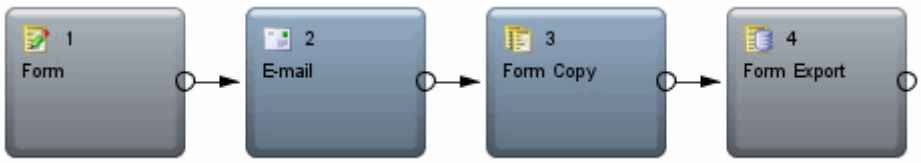
Tasks Overview

A *task* is an action or piece of work that forms one logical step within a process. The tasks that can be used in processes are available from the **Task Library**.

Tasks can be classified in a number of ways:

- Start and End Tasks
- Predecessor and Successor Tasks
- Manual and Automated Tasks
- Synchronous and Asynchronous Tasks
- Parent, Child, and Leaf Tasks

Figure 2 Example tasks



Start and End Tasks

A start task is a task at which the process begins processing. Any task that does not have a predecessor task is a start task. In Figure 2, the start task is the **Form** task.

An end task is the final task in a process. When the end task transitions to a Done state, the process is complete. Any task can serve as an end task.

Predecessor and Successor Tasks

Predecessor tasks are tasks that execute before another task. Predecessor tasks must complete or cancel before their successor tasks can start. In Figure 2, the Form task is the predecessor of the E-mail task.

In most cases, the APA Server does not execute a task until all of its predecessors have been completed or cancelled. It is important to note the behavior of a task with multiple predecessors. A task is not executed repeatedly as each of its predecessors completes. Instead it executes a single time when all of its predecessors are complete.

Successor tasks start after the predecessor tasks complete. In Figure 2, E-mail is the successor of Form. If a task has multiple successor tasks, then the successor tasks are all executed in parallel.

Manual and Automated Tasks

A manual task is a task that is not capable of automation and is therefore outside the scope of APA. An example of a manual task is printing a document and delivering it, by hand, to an individual or department.

APA supports automated tasks that require human or machine resources to support process execution. Where a human resource is required, a task is assigned to a process participant (see “Task Assignment” on page 95). Automated tasks are generally *synchronous* tasks.

You can have both manual and automated tasks within a process. To include a manual task in a process, use the Manual task type. Manual tasks can be included in a process definition, but do not form part of the resulting process. Manual tasks are *asynchronous* tasks.

Synchronous and Asynchronous Tasks

There are two types of tasks in the APA Process Model: Synchronous and Asynchronous. A process can contain both types.

Synchronous tasks are automated tasks that represent the execution of code. When the execution of code is complete, a synchronous task automatically transitions into a Done state (see “Task states” on page 125). Synchronous tasks can invoke other systems locally or remotely, or interact with databases. There is no time limit for the execution of a synchronous task, but they are usually very short-lived because their execution time is only as long as it takes to execute the task's code.

If a task will take an indefinite amount of time, as when human intervention is needed, or when an external system is invoked asynchronously, the task is an asynchronous task. Asynchronous tasks do not transition automatically to a Done state. Upon completion, asynchronous tasks must be explicitly moved into a Done state by a user or external application. An external application must notify the engine that an asynchronous task is complete through an API call.

Parent, Child, and Leaf Tasks

A parent task is a task that contains other (child) tasks. The process template itself is a parent task because it contains all of the tasks in the process. In APA, a process is modeled by a hierarchy of tasks with a single root task at the apex of the hierarchy. The root task may have any number of child tasks which may themselves have any number of children and so on. Container tasks are also parent tasks because they contain their own child tasks.

Leaf tasks are tasks without children. They represent the single units of work involved in the process. Leaf tasks are where the actual work of the process is performed.

Process Control Tasks

Process Control Tasks control how a process is executed.

Container Task

The Container task is a container that can hold other tasks. A Container task can be used to group tasks that constitute some significant part of a process.

Sub-process Task

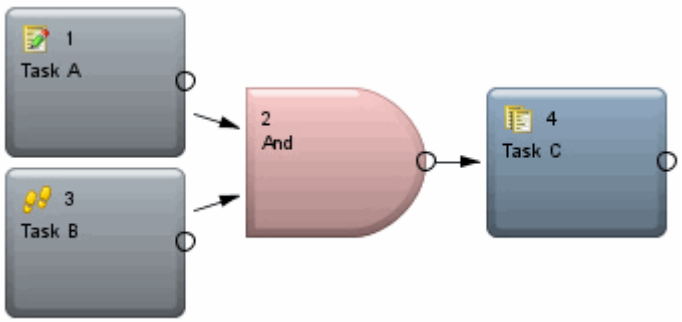
The Sub-process task embeds an existing process (that has been published to the APA Server), in the current process.

CAUTION It is possible to call sub-processes in a recursive or circular manner. However, this could cause a process to go into an endless loop and never finish.

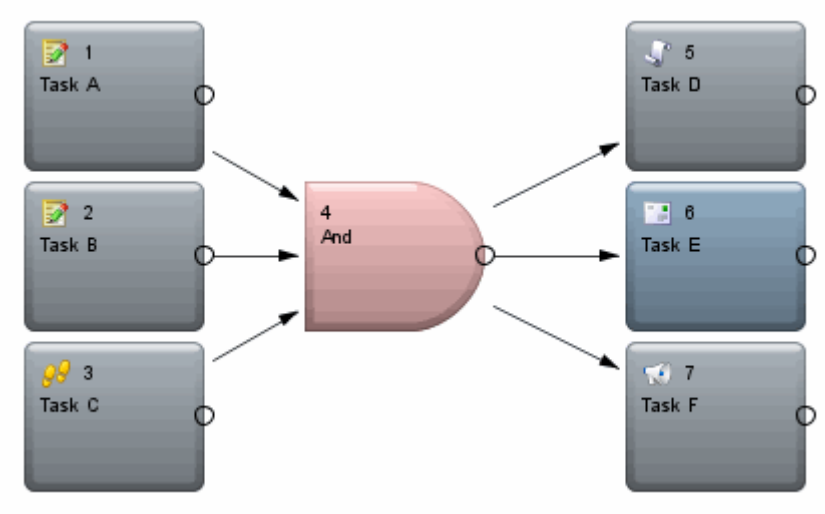
And Task

The And task is used where two or more paths in a process converge. The And task requires all of its predecessor tasks to be completed before the successor task can start.

If multiple tasks are connected to a single successor task, the presence of an And task is implied (it is default behavior that a task does not start until all of its predecessors are complete).

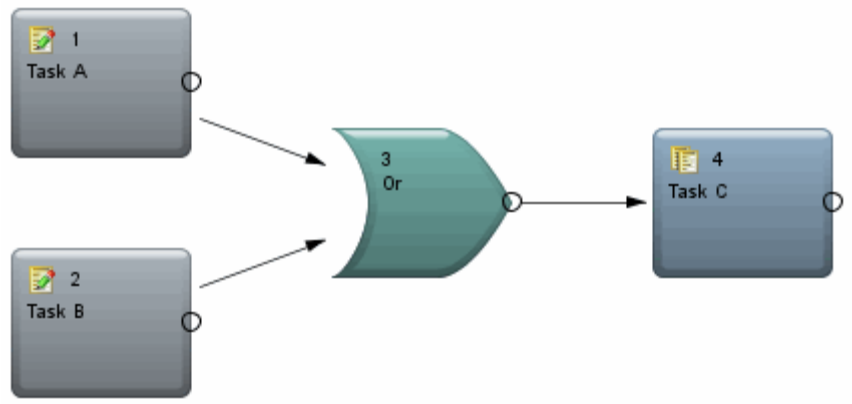


An And task can also be used to reduce the number of connections required to connect multiple tasks to multiple successors.



Or Task

The Or task is used where two or more paths in a process converge. The Or task requires at least one of the predecessor tasks to be complete before the successor task can start.



When the Or task allows the successor task to proceed:

- Any predecessor tasks that are not actively executing are cancelled.
- Predecessor tasks are active can either be cancelled or allowed to complete. To configure this behavior, see “Configure an OR Task” on page 99. If active predecessor tasks are cancelled, and this causes an item to be removed from a user’s APA Inbox, the item is replaced with a message indicating that it was withdrawn.

Branch Task

The Branch task is used to make a decision. The Branch task executes one task from a number of successor tasks. The chosen path can be determined by providing a default path, writing script, or writing an expression. When a branch task follows a form task, the path can also be determined by the submit action chosen on submission of the form.

If you configure a branch task to choose a path using an expression, the following conditions apply:

- If the expression evaluates to false, the branch path “0” is followed.
- If the expression evaluates to true, the branch path “1” is followed.
- If the expression evaluates to a number, the branch path assigned that number is followed.

The default path of a Branch task is used in the following scenarios:

- The Branch task is configured to use the **Default Path**. The default path can be changed dynamically using script.
- The Branch task is configured to use an **Expression** to choose the successor task, but the expression fails.
- The Branch task is configured to use **Another Task** to choose the successor task, but the task fails.

TIP To go back to a previous task in the process, use the Loop task.

Loop Task

The Loop task is used to return the process to an earlier task.

Loop Tasks require a condition that controls whether the loop should continue or terminate. If the condition evaluates to true, the loop continues. If the condition evaluates to false, the loop terminates.

The condition can be a default value, an expression, or the result of another task. The default value can be changed dynamically using script.

Loops can be nested and interleaved. A loop must be connected to a *target* task. The target task is the first task inside the loop. The loop body consists of the target task and its successor tasks, until the loop (or a task downstream from the loop) is reached. The loop does not evaluate until all tasks within the loop body have completed or cancelled.

Gate Task

The Gate task is used to prevent execution of successor tasks based on an arbitrary condition. When the gate is opened, the successor tasks can execute. Gate tasks in process instances initiated on the APA Server can only be opened programmatically. Once open, the gate cannot be closed unless it is part of a Loop body.

To test a Gate task in Process Studio, run the process in Flow Simulation Mode, right-click the task, and click **Open Gate**.

NOTE Gates can be opened at any time. They are not required to be active.

Delay Task

The Delay task delays the execution of the process.

For example, you have a Script task that calls an archive program, but you do not want the Script to run until after midnight when the system backup occurs. You can include a Delay in the process prior to the execution of the Script task that restricts the Script task from executing until after midnight.

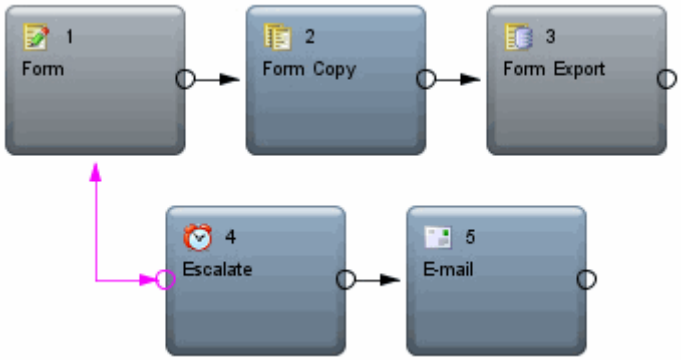
Delay tasks can be configured to use the business calendar (rather than absolute time) when calculating how long the process is delayed. See “Use Business Calendar” on page 117.

Delay tasks can be configured with alternate delay values for use in Process Simulations, to prevent you from having to wait an unreasonable amount of time for the task to complete.

Escalate Task

The Escalate task handles tasks that have not completed within a certain period of time. If the target task does not complete within the time specified, the Escalate path is taken.

In the following example, if the Form task does not complete within the time specified, the E-mail task is executed. An e-mail is sent to the address provided in the E-mail task properties.



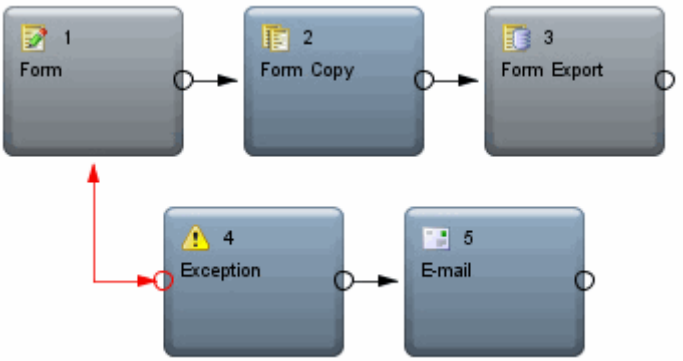
When the Escalate path is followed, the task that did not complete on time remains active. The Escalate task can be used to send an e-mail to the user assigned to the Form task, to remind them to complete it.

Escalate tasks can be configured to use the business calendar for time calculations (rather than absolute time). See “Use Business Calendar” on page 117.

Escalate tasks can be configured with alternate delay values for use in Process Simulations, to prevent you from having to wait an unreasonable amount of time for the task to complete.

Exception Task

The Exception task is used for error handling. The Exception task is linked to another (target) task in the process. If the target task enters an Aborted state, the Exception task is executed. This is useful for implementing script error handling. In the example, if the Form task fails, the Exception path is executed and an E-mail is sent to the address provided in the E-mail task.



Form Tasks

Form tasks are tasks that are associated with forms.

Form Task

The Form task is used to assign a form to a user, either for entering new data or modifying existing data within a process.

Form Copy Task

The Form Copy task is used to assign a read-only copy of a form to a user. This task type can be used to distribute final copies of a form at the completion of a process.

Form Export Task

The Form Export task is used to run all exports defined on the APA Server for a specific form.

NOTE When the Form Export task is executed, the form default values (as defined in Process Studio) are applied for any fields that do not have values.

Form Packet Task

The Form Packet task is used to associate a Packet with a Task. A packet is a collection of forms bound together in a structure for processing. For information about creating Form packets, see “Form Packets” on page 57.

Utility Tasks

Utility tasks are used to perform specific actions.

Script Task

The Script task is used to run Server-Side script. For more information about writing server-side script, refer to the *APA Programming Guide*.

Command Task

The Command task executes an executable file. Substitution syntax can be used to make the command, or its arguments, dynamic. See “Create Dynamic Values Using Substitution Syntax” on page 34.

Message Task

The Message task sends a message to a user’s APA Inbox. The message can be in the form of either a link (URL) or text. The process does not wait for a response from the user to continue. Substitution syntax can be used to make the message dynamic. See “Create Dynamic Values Using Substitution Syntax” on page 34.

E-mail Task

The E-mail task is used to send an e-mail message. Substitution syntax can be used to make the message dynamic. See “Create Dynamic Values Using Substitution Syntax” on page 34.

Manual Task

The Manual task represents a manual step in a process. It can be used to include a task that is not capable of automation and is outside the scope of APA (for example, printing a document and delivering it, by hand, to an individual or department).

When you include a Manual task in a process, the assigned user is notified of the task by a message in their APA Inbox. This message can include a web link or text-based instructions.

IMPORTANT Manual tasks do *not* complete until the assigned user marks the task as complete.

If you choose to send instructions to the assigned user’s APA Inbox, the user is given an option to confirm the task is complete.

APA does not provide a user interface to complete a Manual Task that sends a Web Link. The task must be completed through the APA Process API or SOAP API. If you send a web link to a custom web application, the custom application can complete the task using the API. This requires the application to identify the task. This can be done by passing extra information, such as the task ID, as URL parameters in the web link.

Web Service Task

The Web Service task is essentially a web service client. This task can be used to create a SOAP request by reading a WSDL file, sending the SOAP request to the server and displaying the web service response back to the user. Substitution syntax is supported in the request string. See “Create Dynamic Values Using Substitution Syntax” on page 34.

The Web Service task is not displayed in the **Task Library** until Process Studio has successfully connected to an APA Server that supports the premium workflow option.

Committee Task

The Committee task allows a user to submit a form to be reviewed by committee of approvers.

Criteria can be established for the task to be approved:

- **All approvers must approve**. All members of the committee must approve the form before the task is considered approved.
- **Minimum number to approve**. A minimum number of members must approve the form before the task is considered approved. The number of approvers must be greater than zero and less than or equal to the total number of committee members.
- **Minimum percent to approve**: A minimum percentage of members must approve the form before the task is considered approved. The percentage of approvers must be between zero and one hundred, and must be an integer.

The committee task can also be configured to wait for all form submissions. If this option is enabled, all form submissions must take place before the task can be considered complete, regardless of the approval status.

Other Tasks

The following tasks can help you work more efficiently.

Composite Task

The Composite task allows you to create a new task in which two or more tasks are defined using one task name. The composite task can then be used repeatedly in processes. This is useful in situations where certain tasks are used in the same pattern repeatedly.

For information about how to create a composite task, see “Save a Composite Task” on page 87.

Sticky Note Task

The Sticky note task is used to add comments to process templates. See “Add a Comment to a Process” on page 89.

CHAPTER 6

Add and Connect Tasks

This section describes how to create and connect Tasks in Process Studio.

After creating a process, tasks can be added to the process template. A task is an action or piece of work that forms one logical step within a process. For information about the types of task that are available in Process Studio, see “Tasks Overview” on page 69.

- Add a Task
- Connect Tasks
- Save a Composite Task
- Delete a Task
- Convert a Task to another Type
- Add a Comment to a Process

Add a Task

To add tasks to a process, you must be in Graphical View. If you are in XML View, click the **Graphical View** tab.

To add a task to a process template

- Drag the task from the **Task Library** onto the **Graphical View**.

When you release the mouse button, the task is added to the process template.

Related Topics

- “Tasks Overview” on page 69
- “Connect Tasks” on page 83
- “Configure Tasks” on page 91

Add a Task Pattern

Arrangements of tasks that are commonly used are provided in the Patterns Drawer of the Task Library.

To add a pattern to a process template

1. In the **Task Library**, click the **Patterns** drawer.
2. Drag one of the task patterns onto the **Graphical View**.

   When you release the mouse button, the pattern is added to the process template.

Related Topics

- “Add a Task” on page 81
- “Task Patterns” on page 151

Add a Container Task

A Container task is a container that can hold other tasks. A Container task can be used to group tasks that constitute some significant part of a process.

To add a container task to the process

1. In the **Task Library**, click the **Control** drawer.
2. Drag the **Container** task from the **Task Library** onto the **Graphical View**.

Add Tasks to a Container

To add tasks to a container, use the following procedure.

To add tasks to a container

1. Double-click the Container task on the process template **Graphical View** or in the **Process Tree**. Container tasks are displayed with a folder icon in the process tree.

The process template **Graphical View** displays the contents of the Container.

2. Drag tasks from the **Task Library** onto the **Graphical View**.

NOTE The number of tasks in a Container is displayed in parenthesis on the Container Task in **Graphical View**.

NOTE If you hover the mouse over the Container task, the tasks it contains are displayed in a tooltip.

3. To move back to the main process template:
   - Double-click the process name in the **Process Tree**
   - On the **View** menu, click **Parent Level**
   - Click **Parent Level** .

Related Topics

- “Save a Composite Task” on page 87

Connect Tasks

After tasks are added to a process template, they need to be connected. The connections between tasks define the order in which the tasks are executed.

- A *connector* is a small circle that appears on the edge of a task in the **Graphical View**. It is used to connect the task to other tasks in the process.
- A *path* is the line used to connect tasks.
- A process can have multiple paths between its start and end tasks.

Various types of connectors and paths are used depending on the type of task. See the following topics:

- “Connect a Branch Task” on page 84
- “Connect a Loop Task” on page 85
- “Connect an Exception Task” on page 85
- “Connect an Escalate Task” on page 86

To connect other types of task, use the following procedure.

To connect tasks

1. Drag a connector from a task to its successor task.

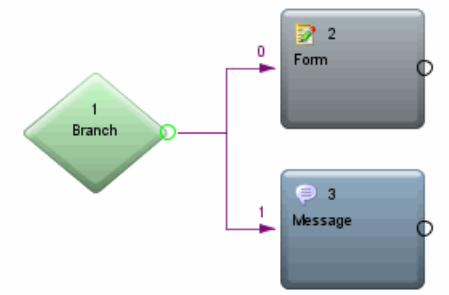
When you release the mouse button, the path is displayed.

Related Topics

- “Add a Task” on page 81
- “Configure Tasks” on page 91

Connect a Branch Task

Branch tasks can be connected to more than one successor task. Only one of these successor tasks is executed.



The path selected by a Branch task is controlled by defining a default path, writing script, writing an expression, or based on another task. Paths are identified by index values.

To connect a Branch task

1. Drag the green connector from the right side of the Branch task to a successor task.

   When you release the mouse button, a path is displayed.

2. Continue this process until the Branch task has been connected to all of its successor tasks.

   Each path is assigned a numeric value (starting with "0"). These values can be edited at any time (see “Configure a Branch Task” on page 100).

NOTE A path index, *p*, is only valid if: *p*>=0 and *p*<*nPaths*-1, where *nPaths* is the total number of path choices for the Branch task.

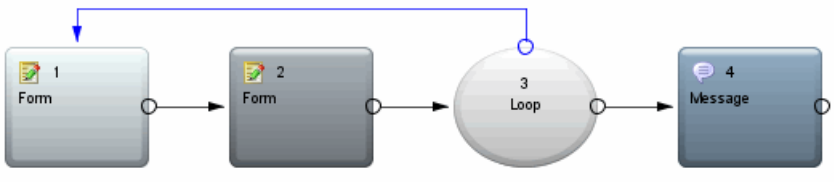
NOTE If an invalid path index is specified, an exception occurs and the task is moved to the Aborted state.

Related Topics

- “Branch Task” on page 74

Connect a Loop Task

Loop tasks can be connected to more than one task and have two connectors.



To connect a loop task

1. Drag the blue connector on the top of the Loop task to the *target* task (the first task inside the loop).
2. Drag the black connector on the right of the loop to the task that should be executed when the loop terminates.

Related Topics

- “Loop Task” on page 74

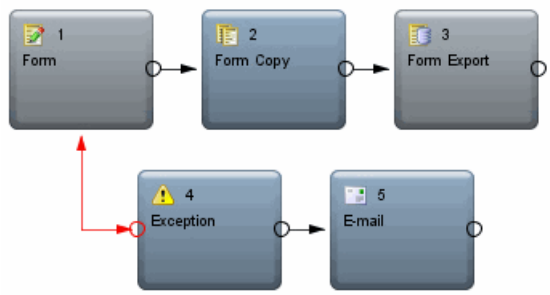
Connect an Exception Task

An Exception task is connected to a *target* task. The Exception task is only executed if the target task enters the Aborted state. To connect an Exception task, you must draw the path from the Exception task towards the target task.

To connect an Exception task

- Drag the red connector from the left side of the exception task onto the target task.

TIP If the path is drawn correctly, it is displayed in red with arrows pointing in both directions.



Related Topics

- “Exception Task” on page 76

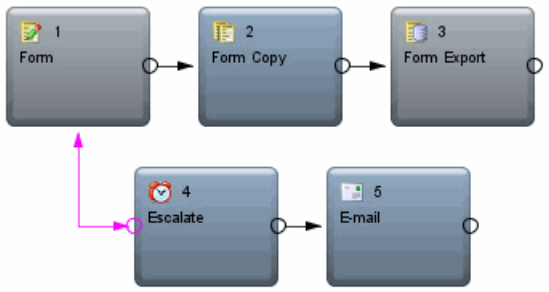
Connect an Escalate Task

An Escalate task is connected to a *target* task. The Escalate task is only executed if the target task is not completed within a specified amount of time. To connect an Escalate task, you must draw the path from the escalate task towards the target task.

To connect an Escalate task

- Drag the pink connector from the left side of the escalate task onto the target task.

TIP If the path is drawn correctly, it is displayed in pink with arrows pointing in both directions.



Related Topics

- “Escalate Task” on page 75

Remove a Connection

To delete a connection between two tasks, use the following procedure.

To remove a connection

1. Click the connection.
2. Press DELETE.

Related Topics

- “Connect Tasks” on page 83

Save a Composite Task

If you regularly use a group of tasks in Process Studio, you can save them as a Composite task to be used in future processes. The Composite task is a new task in which tasks are defined under a single name.

To create a composite task

1. Add the tasks you want to combine to the **Graphical View**. For example, your company performs the same tasks every time a form is approved by the Accounting Department. Add these tasks to the process template and connect them.
2. Select the tasks by dragging the bounding outline around the tasks.
3. Copy the tasks to the clipboard:
   - On the **Edit** menu, click **Copy**
   - Click **Copy**
   - Right-click the selected tasks and click **Copy**
   - Press CTRL + C
4. On the **Tools** menu, click **Options**.

   The **Options** dialog box opens.
5. Click the **Gallery** tab.
6. In the tasks list, click the **Other** tab.
7. Drag **Composite** to the drawer where you want to locate the new Composite task (for example the **Favorites** drawer).

   The **Composite Task Configuration** dialog box opens.
8. In the **Label** box, type a name for the Composite task.

9. In the **Icon** box you can specify a path for an icon to represent the task.
10. Click **Paste from clipboard**.

    The XML for the tasks that you copied is pasted into the **XML** box.
11. Click **OK**.

    The Composite task appears in the **Task Library** and can be used to create future processes.

Related Topics

- “Add a Task” on page 81

Delete a Task

To delete a task or object from the process template, use the **Delete** command.

NOTE You cannot delete an object from an Instance. Tasks must be deleted from the Process Template.

To delete an object

1. Select the object.
2. Use the delete command:
   - On the **Edit** menu, click **Delete**.
   - Click **Delete** .
   - Press DELETE.
   - Right-click an object and click **Delete**.

Related Topics

- “Add a Task” on page 81
- “Convert a Task to another Type” on page 88

Convert a Task to another Type

To change the type of a task, use the **Change Type** command.

NOTE You cannot change the type of a task in an Instance. The type of a task can only be modified from the Process Template.

To change the type of a task

1. Select the Task you want to change.
2. On the **Edit** menu, click **Change Type**.

   The **Change Type** dialog box opens.
3. In the **New Type** list, click a new type.
4. Click **OK**.

Related Topics

- “Add a Task” on page 81
- “Delete a Task” on page 88

Add a Comment to a Process

Comments can be added to a Process using the Sticky Note task. Comments can be helpful if the Process needs to be modified in the future.

To add a Sticky Note to a process template

1. In the **Task Library**, click the **Other** drawer.
2. Drag the Sticky Note task from the **Task Library** onto the **Graphical View**.

   A note is added to the Process Template.
3. Click the sticky note to select it.
4. Wait a few seconds, and then click the note again.

   You can now edit the text contained in the note.
5. Type text in the note.
6. When you have finished, click an empty part of the **Graphical View**.

Change the Color of a Sticky Note

To change the color of a sticky note, use the following procedure.

To set the color of the Sticky Note

1. Double-click the Sticky Note.

   The **Note Color** dialog box opens.

2. Select a color.
   - In the **Swatches** tab, click a color.
   - Alternatively, click the **HSB** tab or **RGB** tab and define a custom color.
3. Click **OK**.

CHAPTER 7

Configure Tasks

This section describes how to configure tasks.

- Configure Tasks
- Task Assignment
- Configure Task-specific Properties
- Task Properties

Configure Tasks

Each task has associated properties that define its behavior. These properties vary based on the selected task.

TIP Task Properties can be modified in either Process Templates or Instances. Changes made in an Instance do not affect the Process Template.

To view or configure Task Properties

1. To view or configure task properties:
   - Select the task. On the **Edit** menu, click **Task Properties**.
   - Select the task and press ALT + ENTER.
   - Double-click the task (except container tasks).

- Right-click the task and click **Task Properties**.

The **Task Properties** dialog box opens.

2. View or configure the properties on the following tabs (if applicable to the selected task):

General	General properties identify a task within the process, estimate its duration, and enable or disable script. Comments can also be added to the task for future reference. See “Configure General Properties” on page 92.
Assignment	Some tasks must be assigned to users. See “Task Assignment” on page 95 and “Configure Task Assignment” on page 97.
Attributes	To assign attributes to a task, see “Configure Attributes” on page 94.
Simulation	Process Studio includes features for simulating how a process might behave when placed under load. Before you can run a load simulation, some tasks must be configured. See “Configure Tasks for Load Simulation” on page 127. For more information about process simulations, see “Simulate a Process” on page 123.

These properties are the same for all types of tasks.

3. Configure task-specific properties. See “Configure Task-specific Properties” on page 98.
4. Click **OK**.

Related Topics

- “Add a Task” on page 81
- “Connect Tasks” on page 83

Configure General Properties

General properties identify a task within the process, estimate its duration, and enable or disable script. Comments can also be added to the task for future reference.

Name

The `Name` property specifies a name for the task. Maximum 50 characters (alpha-numeric, dash, underscore, and spaces).

To rename a task

- In the **Name** box, type a name for the task.

Duration

The `Duration` property is an estimate of the time required to complete the task. The value is formatted as `DAY:HOUR:MINUTE:SECOND.MILLISECOND`.

The `Duration` property is used for process simulation. After a process is published to the APA Server, you can analyze the process to see how the estimated duration compares to the actual time taken for the task to complete. For more information, refer to the *Management Console User Guide*.

NOTE The Duration of a Container task is set automatically using the durations of the tasks it contains. The Duration of a Container task is set to the greater value of:

- The total duration required for any connected tasks.
- The longest duration of any task that is not connected.

For example: A Container task contains two tasks, Form and Manual. You assign a Duration of 1 day to the Form task and 2 days to the Manual task.

If the Form and Manual tasks are connected, the Container task has a Duration of 3 days. If the Form and Manual tasks are not connected, the Container task has a Duration of 2 days.

To estimate the duration of a task

- In the **Duration** box, type or select a duration.

Scoped ID

The `Scoped ID` is a number that identifies a task within the process. Each task in a process has a unique Scoped ID.

To edit the Scoped ID

1. Click **Edit**.

   The **Edit Scoped ID** dialog box opens.

2. Type a new Scoped ID. The Scoped ID must be a valid 32-bit signed integer.
3. Click **OK**.

Design Comments

The Design Comments property allows the process author to type comments about the task for future reference. Comments are especially useful if other designers need to edit the process template.

TIP The Name, Duration, and Design Comments properties are displayed in a tooltip when you hover the mouse pointer over a Task in a Template or Instance.

To edit the Design Comments

- In the **Design Comments** box, type comments about the function of the task.

Enable Script

To enable script for this task, select the **Enable Script** check box. Click **Edit** to open the **Script Editor**. For information about the script editor, see “Create Custom Scripts” on page 37.

To enable script in Flow Simulation Mode, select the **Enable Script in Simulation** check box. (This option is only available if the **Enable Script** check box is selected).

NOTE Deprecated script is not supported in Flow Simulation Mode. Additionally, some API objects are not supported in Flow Simulation Mode due to the fact that the complete server environment is not available.

Configure Attributes

You can apply attributes to Tasks in a process. For example, you can create an attribute that counts the number of times a Loop task is iterated.

Add an Attribute

To create a new attribute, use the following procedure.

To add an attribute

1. Click **New**.

   A new item is added to the list.

2. In the **Name** and **Value** cells, type the name and value of the attribute. Double-click a cell to edit its value.
3. In the **Type** cell, click a data type for the attribute: **Text**, **Number**, or **Scoped ID**. A Scoped ID is an identifier that is unique within a process.

Delete an Attribute

To delete an attribute, use the following procedure.

To delete an attribute

1. Click the attribute in the list.
2. Click **Delete**.

Task Assignment

Form, Form Copy, Message, and Manual Tasks must be assigned to users. The assigned user is the person who is required to complete the action or work required by the task.

Users must also be assigned to Committee tasks (as members of the committee).

A task can be assigned to a user directly or indirectly (for example through a work queue or relationship).

Process Originator

A task can be assigned to the *Process Originator*. This is the user who initializes the process in the Portal or Web Desktop.

User or Work Queue

A task can be assigned to a specific user or work queue.

A work queue is a group of users that can be assigned tasks. A work queue has a common inbox and any member of the group can complete the tasks assigned to the queue. Work queues are configured using the Management Console.

Performer of Another Task

A task can be assigned to a user who performs another task in the process.

Relationship

Relationships describe the connections between users in an organization.

Tasks can be assigned to a relationship. For example, a form can be routed to a user’s manager (rather than a specific user), and the form is routed to the correct manager regardless of the user’s identity. Processes that assign events based on relationships are also easier to maintain because they are not affected by changes in an organization's hierarchy.

Relationships are configured in the Management Console. For information about configuring relationships, refer to the *Management Console User Guide*.

Relationship - Process Originator

A task can be assigned to a user who is related to the process originator. The Process Originator is the individual who initialized the process on the APA Server.

For example, if the Relationship selected is *Manager* and the process originator is John Smith, the task is assigned to John Smith's Manager.

Relationship - User or Group

A task can be assigned to the selected Relationship of a User or group.

For example, if the Relationship selected is *Manager* and you have selected the user John Smith, the task is assigned to John Smith's Manager.

Relationship - Performer of Another Task

A task can be assigned to a user who is related to the performer of another task in the process.

For example, if the Relationship selected is *Manager* and the Performer of the selected task is John Smith, the task is assigned to John Smith's Manager.

None

A task can be assigned no users. This might be appropriate if you intend to set the assigned user for the task at a later time (for example, using script).

Assignment Priority

Tasks can be assigned a priority (low, normal, or high).

When a task is assigned to a user, the Priority flag is displayed in the **Importance** column next to the assigned form in the user’s inbox. For more information about the **Importance** column, refer to the *APA Portal online help* or *Web Desktop online help*. Additionally, when a user clicks **Get Next Item** in a Work Queue, the higher priority tasks are displayed before the lower priority tasks.

Abort if Unassigned

Tasks can be configured to enter the Abort state once activated, if they do not have an assignment.

If the task does not enter the Abort state, it remains active but not assigned. Unassigned tasks can be reassigned using the **Process Activity** page in the Management Console.

Configure Task Assignment

To configure task assignment, use the following procedure.

To configure task assignment

1. Click the **Assignment** tab.
2. Click the appropriate assignment. For more information about how tasks can be assigned, see “Task Assignment” on page 95.
   - **Process Originator**
   - **User or Work Queue**
   - **Performer of Another Task**
   - **Relationship**
   - **None**
3. If you assign the form to a Relationship, complete the following steps.
   a. click **Browse**.

      The **Search Users** dialog box opens.

   b. Click the relationship that you want to use, and click **OK**.

   c. Click one of the relationship types:
      - **Process Originator**
      - **User or Group**
      - **Performer of Another Task**

      You might also need to select a user, work queue or group. Click **Browse** to search the APA server. For information about how to search the APA server, see “Search the APA Server” on page 132.
4. In the **Assignment Priority** list, click a priority: **Low**, **Medium**, or **High**.
5. If you want the task to enter the *Abort* state once activated (if it is not assigned to a user), select the **Abort if Unassigned** check box. Clear this check box if you want the task to remain active but unassigned.

6. Click **OK**.

Related Topics

- "Task Assignment" on page 95

Configure Task-specific Properties

This section describes how to configure the properties that are specific to each type of process task.

- Configure a Sub Process Task
- Configure an OR Task
- Configure a Branch Task
- Configure a Loop Task
- Configure a Delay Task
- Configure an Escalate Task
- Configure a Script Task
- Configure a Command Task
- Configure a Message Task
- Configure a Committee Task
- Configure an E-mail Task
- Configure a Manual Task
- Configure a Web Service Task
- Configure a Form Task
- Configure a Form Copy Task
- Configure a Form Export Task
- Configure a Form Packet Task

Configure a Sub Process Task

The sub-process task embeds an existing process (that has been published to the APA Server), in the current process.

To configure a Sub Process task

1. Open the **Sub Process Task Properties** dialog box and click the **Sub Process** tab.
2. Click **Search**.

   The **Search Process Template** dialog box opens.
3. Browse to or search for the process template you want to use in the sub-process. Click the process and then click **OK**. For information about how to search the APA server, see "Search the APA Server" on page 132.

NOTE The processes that you can select might be restricted by permission settings.

4. Click one of the following options to select how the process template data is shared between the parent process and the sub-process:
   - **Embed Sub-process Instance in Main Process**. The parent process and sub-process share the same data set. Data changes made in either process are automatically reflected in the other process. The main process must wait for the sub-process to complete.

     In the Management Console, the tasks in the sub-process are displayed as if the sub-process were a container task.
   - **Launch Sub-process Instance as New Top-level Process**. The sub-process is launched as a new process on the APA server. In the Management Console, there is no way to view the tasks in the sub-process from the main process. The sub-process is displayed as a separate process running on the APA server.
     - **Initialize Sub-process Instance with Main Process Data**. Select this check box to initialize the sub-process with attributes from the parent process. Any attribute changes made in either process after the sub-process is initiated are *not* reflected in the other process.
     - **Wait for Sub-process Instance to Complete**. Select this check box to mark the sub-process task as complete only when the sub process has finished. Otherwise, the sub-process task is marked as complete as soon as the sub-process is initiated.
5. Configure the simulation options:
   - Select the **Invoke in Simulation** check box to execute the sub-process in Flow Simulation Mode. If the **Invoke in Simulation** check box is selected, you must select a process template from the **Process Template** list. The process template must be open in Process Studio.

Configure an OR Task

The OR task is a process control task that requires at least one of its predecessor tasks to be complete before its successor task can start.

To configure an Or Task:

1. Open the **OR Properties** dialog box and click the **OR** tab.
2. Select or clear the **Cancel unfinished predecessors as soon as any predecessor is done** check box.

   If this check box is selected, the process continues once any predecessor task is complete. All other predecessor tasks are cancelled.

   If this check box is cleared, the process continues once any predecessor is complete. However, all predecessor tasks that are not complete remain active.

Configure a Branch Task

The Branch task is used to make a decision. The Branch task executes one task from a number of successor tasks.

To configure a Branch task

1. Open the **Branch Task Properties** dialog box and click the **Branch** tab.
2. The **Branch Paths** area is used to define the branches. If the branch task has been connected to successor tasks, branches already exist and are displayed in the list.
   - To insert a new branch, click **New**. A new condition is added to the **Branch Paths** table. In the **Task** column, click a task to assign to the branch.

NOTE It is possible to set up a Branch Path that does not have an assigned Task. If this path is executed, the Branch task transitions to *Done* and all Branch targets become passive.

   - To remove a branch, click the branch you want to remove and click **Delete**. This causes the index numbers to change so that they remain zero-based and contiguous.
   - To change the order of branches, click the branch you want to move and click **Move Up** or **Move Down**. The default branch remains the same as before the move operation.
   - To define a default path, select the relevant check box in the **Default** column.
3. In the **Choose Path Based On** area, click one of the options to choose how the branch task should choose its successor task:

Default Path	The default path is selected.
Expression	The path is selected based on an expression. Click **Edit** to open the Expression Builder. ■ If the expression evaluates to false, the branch path “0” is followed. ■ If the expression evaluates to true, the branch path “1” is followed. ■ If the expression evaluates to a number, the branch path assigned that number is followed. For more information about the Expression builder, see “Build Expressions” on page 32. For information about the Expression Language, see “APA Expression Language” on page 171.
Another Task	The path is selected based on the submit action chosen when a form is submitted. See “Use Another Task to Determine Path Selection” on page 101.

4. Type or select a number in the **Select an optional alternate path if the chosen path does not exist** box. If the value returned is greater than the highest branch index, the alternate path is used. This might occur with processes originally created prior to version 5.0 of APA.

Use Another Task to Determine Path Selection

A branch task can be configured to select a process path based on the submit action chosen when a form is submitted.

Before you can set up a Branch task to execute based on **Another Task**, you must configure the Form task that precedes the Branch task. You must also make sure there are tasks available in your process toward which you plan to point your Branch paths.

To set up Another Task on a Branch Task

1. In the **Choose Path Based On** area, click **Another task**.
2. In the **Another Task** list, click a Form Task. The Branch Path is chosen based on the submit action selected when this form is submitted.
3. Click **Edit Mapping**.

   The **Branch Mapping** dialog box opens.
4. To add a condition to map a task output to a branch path, click **New**.

   A new condition is added to the list.

a. In the **Task Output** cell, click a task output.

NOTE Custom submit actions can be used. You must select the value of the custom submit action in the **Task Output** box. Only Submit or Reject submit action types can be selected.

b. In the **Branch Path** cell, type or select the branch path that should be chosen if the submit action in the **Task Output** column is selected.

5. To delete a submit action from the **Task Output** list, click the row and click **Delete**.
6. Click **OK**.

Configure a Loop Task

The Loop task is used to return the process to an earlier task.

To configure a Loop task

1. Open the **Loop Task Properties** dialog box and click the **Loop** tab.
2. Click **True** or **False** to select a default value. A default value of True indicates that the loop should continue. A default value of False indicates that the loop should terminate.
3. In the **Follow Path Based On** area, click one of the options to choose how the loop is controlled.

Default Value or Script	A default value determines whether the loop continues or terminates. The default value can be changed using script.
Expression	An expression determines whether the loop continues or terminates. Click **Edit** to open the Expression Builder. For more information about the Expression builder, see “Build Expressions” on page 32. For information about the Expression Language, see “APA Expression Language” on page 171.
Another Task	The submit action of a submitted form determines whether the loop continues or terminates. See “Use Another Task to Control Loop Execution” on page 103.

Use Another Task to Control Loop Execution

A loop can be configured to execute or terminate based on the submit action chosen when a form is submitted.

Before you can set up a Loop task to execute based on Another Task, you must configure the Form task that precedes the Loop task. You must also make sure there are tasks available in your process toward which you plan to point your Loop.

To set up Another Task on a Loop Task

1. In the **Choose Path Based On** area, click **Another task**.
2. In the **Another Task** list, click a Form Task.
3. Click **Edit Mapping**.

   The **Loop Mapping** dialog box opens.

   If any of the submit actions in the **Task Output** list are chosen, the loop continues. If none of these actions are chosen, the loop terminates.
4. To add a new task output to the list, click **New**.

NOTE Custom submit actions can be used. You must type the name of the custom submit action in the **Task Output** box. Only Submit or Reject submit action types can be selected.

5. To delete a task output from the list, click the row and click **Delete**.
6. Click **OK**.

Configure a Delay Task

The Delay task delays the execution of the process.

To configure a Delay task

1. Open the **Delay Task Properties** dialog box and click the **Delay** tab.
2. In the **Delay** box, type or select the amount of time that the process is delayed.

NOTE The format of the value `0:00:00:00.000` is DAY:HOUR:MINUTE:SECOND.MILLISECOND.

3. Select the **Use business calendar** check box to use business time as defined by the business calendar. Clear the check box to specify a value as an absolute time. For more information about the Business Calendar, see “Use Business Calendar” on page 117.
4. To specify an alternate delay value for use in Process Simulations, select the **Use Alternate Delay in Simulation** check box and type or select a shorter delay value (for example, a few seconds).

Configure an Escalate Task

The Escalate task provides a solution for handling tasks that have not competed within a certain period of time.

To configure an Escalate task

1. Open the **Escalate Task Properties** dialog box and click the **Escalate** tab.
2. In the **Delay until escalation** box, type or select the amount of time that should pass before the Escalate task is used.

NOTE The format of the value `0:00:00:00.000` is DAY:HOUR:MINUTE:SECOND.MILLISECOND.

3. Select the **Use Business Calendar** check box to interpret the delay value using the business calendar, rather than as an absolute value. For more information about the Business Calendar, see “Use Business Calendar” on page 117.
4. To specify an alternate delay value for use in Process Simulations, select the **Use Alternate Delay in Simulation** check box and type or select a shorter delay value (for example, a few seconds).

Configure a Script Task

The Script task is used to run Server-Side script.

To configure the Script task:

1. Open the **Script Properties** dialog box and click the **Script** tab.
2. Select the **Wait for Notification** check box if you want the task to wait until an external application notifies APA before the script task transitions to the *Done* state. The task can be set to the *Done* state programmatically through the Process API. Refer to the *APA Programming Guide* for additional information.

If the **Wait for Notification box** is cleared, the task is marked complete as soon as the script method returns.

Configure a Command Task

The Command task executes an executable file.

To configure a command task

1. Open the **Command Task Properties** dialog box and click the **Command** tab.
2. Click **File**.

   The **Select** dialog box opens.
3. Browse to the file you want to execute. Click the file and then click **Select**.

NOTE The file must be available to the APA Server when the process is running.

4. Configure any parameters.

   For example, if you have selected `ACME_ReportGenerator.exe` as the executable command, a parameter could be the directory in which the report is saved: `C:/ReportGenerator/OutputDirectory/`.

   - To define a new parameter, click **New**.
   - To modify the parameter selected in the Parameters table, click **Edit**.
   - To delete the parameter selected in the Parameters table, click **Delete**.
5. To make the process wait for the executable file to complete before, select the **Block process execution until command completes** check box. If this check box is cleared, the task executes the command and the process continues immediately.
6. If you want to run the command in Flow Simulation Mode, check the **Invoke in Simulation** check box.

Configure a Message Task

The Message task delivers a link or message to the APA inbox of the assigned user. For information on how to assign a task, see “Configure Task Assignment” on page 97.

To configure a Message task

1. Open the **Message Properties** dialog box and click the **Message** tab.
2. Click one of the message formats. You can send a Web Link or a Text message to the assigned user's APA Inbox. Substitution syntax can be used in these properties. See “Create Dynamic Values Using Substitution Syntax” on page 34.
   - To send a link, click **Web Link** and type a URL in the box.
   - To send a text message, click **Text** and type a message (for example, instructions for the next step in the process) in the box.
3. Select or clear the **Allow Viewing of Progress Map** check box. See “Allow Viewing of Progress Map” on page 121.

Configure a Committee Task

The Committee task allows an action to be approved by a specified set of approvers.

To configure a Committee Task

1. Open the **Committee Properties** dialog box and click the **Committee** tab.

   The **Members** area is used to define the members of the committee.
2. To add a member to the committee, click **Add**.

   The **Add member** dialog box opens.

   a. Select a user to add to the committee.
      - To add the Process Originator, click **Process Originator**.
      - To add a specific user or work queue, click **User or Work Queue**, and click **Browse**.
      - To add the performer of another task to the committee, click **Performer of Another Task** and click a task in the list.
      - To define a committee member using a relationship, click **Relationship** and then click one of the relationship types.

   b. Click **OK**.
3. To add multiple users to the committee in a single operation, click **Add Multiple**.

   The **Search Users** dialog box opens. See “Search the APA Server” on page 132.
4. To edit an existing committee member, click a name in the **Members** list and click **Edit**.

5. To remove an existing committee member, click a name in the **Members** list and click **Remove**.
6. In the **Criteria** area, click one of the approval criteria. If you click **Minimum number to approve** or **Minimum percent to approve**, you must also type a value.
7. Click the **Form** tab and configure the Form that is submitted to the committee. For more information about form options, see "Configure a Form Task" on page 113.

Configure an E-mail Task

The E-mail task is used to send an e-mail message.

To configure an E-mail Task

1. Open the **E-mail Properties** dialog box and click the **E-mail** tab.
2. In the **From** box, type an e-mail address. This is the address the e-mail will appear to be sent from.
3. In the **To** box, type an e-mail address. The e-mail is sent to this address.
4. If required, type e-mail addresses in the **Cc** and **Bcc** boxes.
5. In the **Subject** box, type a subject for the e-mail.
6. Type the e-mail message.

TIP You can provide multiple e-mail addresses in the **To**, **Cc**, and **Bcc** boxes. Separate the addresses using commas.

TIP You can use substitution syntax to create dynamic values. See "Create Dynamic Values Using Substitution Syntax" on page 34.

Configure a Manual Task

The Manual task represents a manual step in a process.

To configure a Message task

1. Open the **Manual Properties** dialog box and click the **Manual** tab.
2. Configure the message sent to the assigned user's APA inbox:
   - To send the user a web link, click **Web Link** and type a URL in the box. Substitution syntax can be used. See "Create Dynamic Values Using Substitution Syntax" on page 34.

- To send the user text instructions, click **Instructions** and type a message in the box. Substitution syntax can be used. See “Create Dynamic Values Using Substitution Syntax” on page 34.

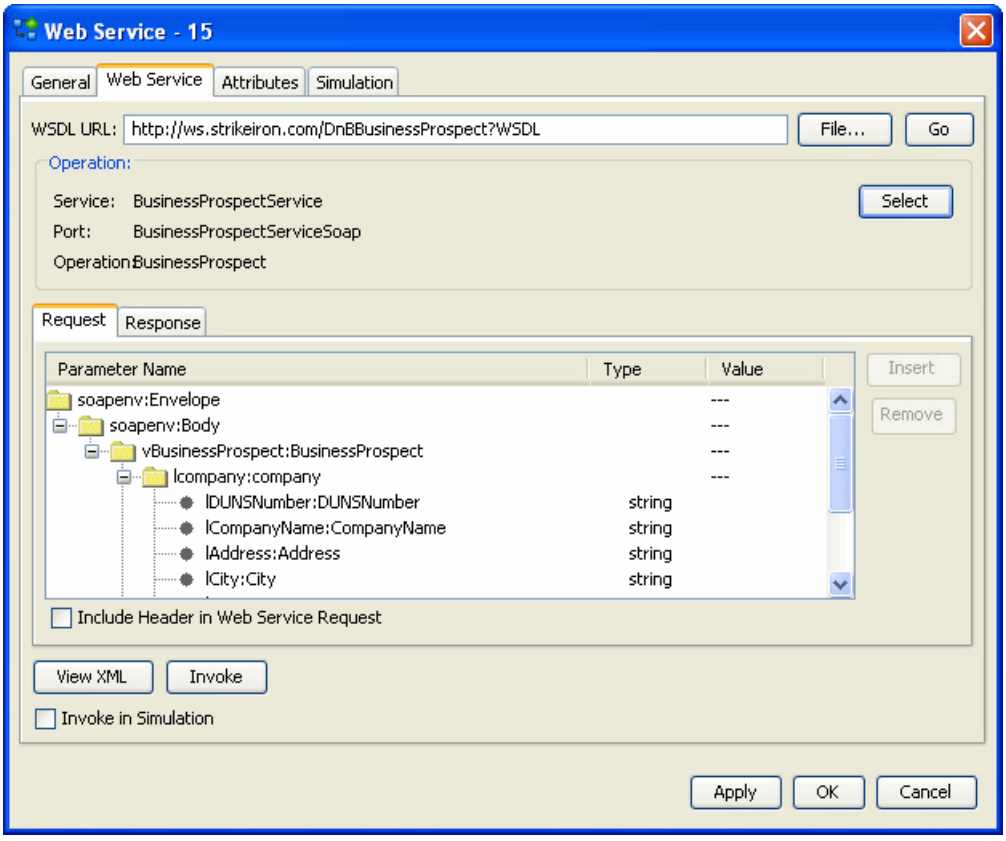
3. Select or clear the **Allow Viewing of Progress Map** check box. For more information, see “Allow Viewing of Progress Map” on page 121.
4. Select or clear the **Allow Attachments to be Deleted** check box. For more information, see “Allow Attachments to be Deleted” on page 118.

Configure a Web Service Task

The Web Service task is essentially a web service client. This task can be used to create a SOAP request by reading a WSDL file, sending the SOAP request to the server and displaying the web service response back to the user.

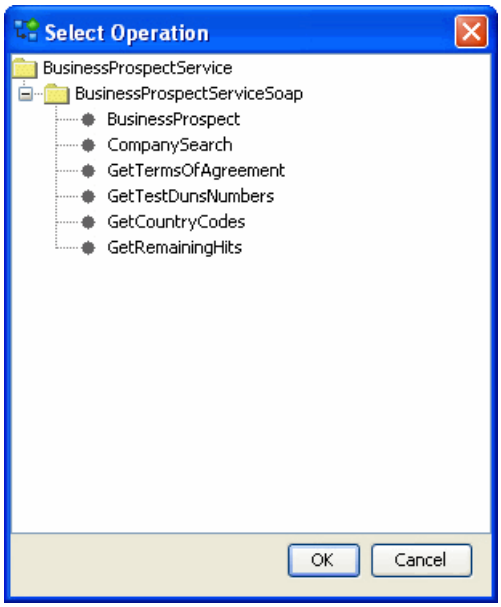
To configure the Web Service task:

1. Open the **Web Service** dialog box and click the **Web service** tab.



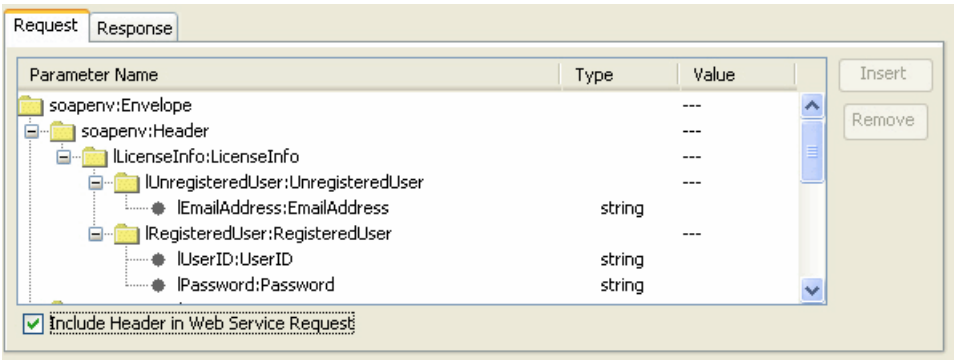
2. In the **WSDL URL** box, type or select the WSDL (Web Services Description Language) file location (a URL or local path). Alternatively, click **File** to browse for a file.
3. Click **Go**.
4. In the Operation area, click **Select**.

   The **Select Operation** dialog box opens. This dialog box displays all the SOAP operations listed in the WSDL.

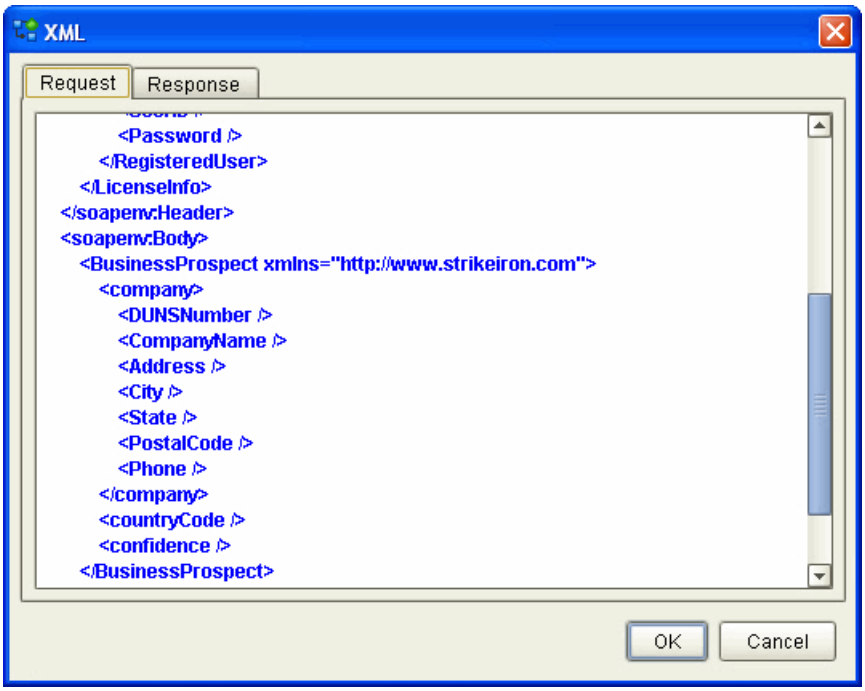


5. In the list, click an Operation.
6. Click **OK**.

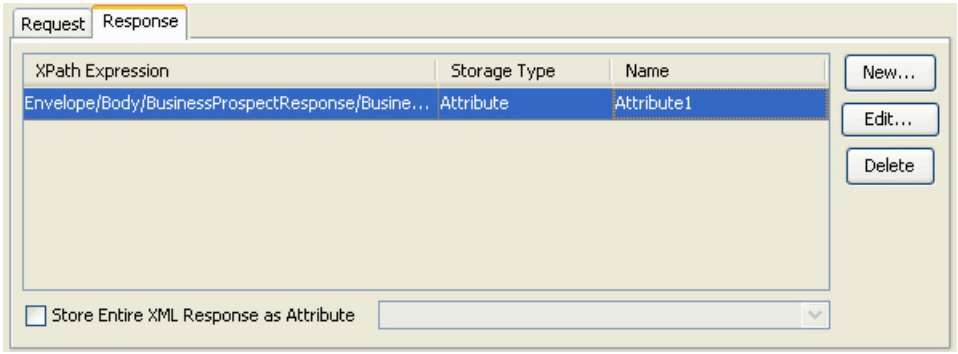
   The **Service**, **Port**, and **Operation** values are displayed. The **Request** tab is also populated. The request refers to the SOAP request, input parameter type and value, based on the XML schema defined in the WSDL URL for the selected operation.



7. The **Value** column in the **Request** tab can be modified. Double-click the **Value** cell in the applicable row to modify the value. Substitution syntax can be used to create a dynamic value. See “Create Dynamic Values Using Substitution Syntax” on page 34.
8. To include a header in the SOAP request, select the **Include Header in Web Service Request** check box. The header is displayed as a part of the SOAP request in the **Request** tab. If the WSDL defines a SOAP header, it can be included as a part of the SOAP request.
9. To send the request to the server and retrieve the response, click **Invoke**.
10. To view the response in XML format, click **View XML**.



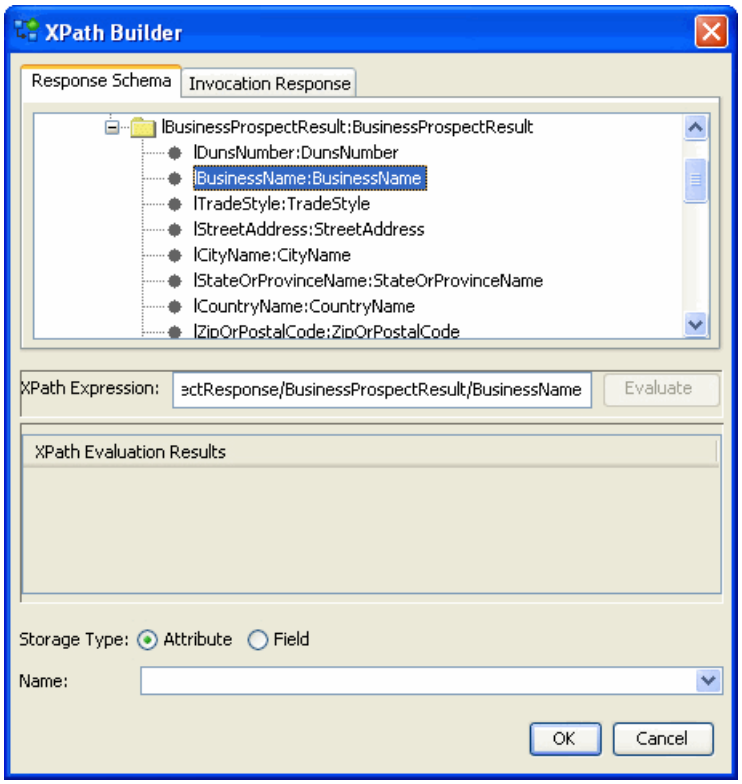
11. Select the **Invoke in Simulation** check box to execute the Web service task during Flow Simulations.
12. To modify array/list handling, click an array in the **Request** tab.
    - To add additional children (up to the amount specified by the WSDL), click **Insert**. This also allows children to be inserted at arbitrary locations within the list by selecting a child node and clicking **Insert**. This will cause the new node to be inserted beneath the selected node.
    - Click **Remove** to remove the selected rows.
13. To load the web service results into a process attribute or field, click the **Response** tab.



The result of the expressions in the **XPath Expression** column load into the process attribute or field specified in the **Name** column.

To create a new expression, click **New.** To modify an existing expression, click **Edit**.

The **Xpath Builder** dialog box opens.



- The **Response Schema** tab displays a tree of the XML response as described in the WSDL. The **Invocation Response** tab displays a tree of the last XML response from an earlier invocation.
- The **Xpath Expression** box displays the XPath Expression. This text area is automatically populated by selecting a node in the tree. Xpath Query Language is supported for selectively choosing the values from the SOAP response
- The **Xpath Evaluation Results** box displays the results of the Xpath Expression.
- The results of the Xpath Expression can be saved to either a process attribute or field. Click **Attribute** or **Field**. The Name box is pre-populated with the list of process attributes or fields for the process. If you type a process attribute or field name that does not exist, a new one is created during the execution of the process.

Click **OK** to close the XPath Builder.

14. To save the web service response to a process attribute, select the **Store Entire XML Response As Attribute** check box.

Configure a Form Task

The Form task is used to assign a form for entering new data or modifying existing data within a process.

To configure a Form task

1. Open the **Form Properties** dialog box and click the **Form** tab.
2. To select a form, click **Search**

   The **Search Forms** dialog box opens.
3. Browse to or search for the form you want to associate with the task. Click the form, and then click **OK**. For information about how to search the APA server, see “Search the APA Server” on page 132.
4. Configure the form options:

Allow Form to be Edited	See “Allow Form to be Edited” on page 117.
Set Field Properties	See “Set Field Properties” on page 117
Allow Attachments to be Deleted	See “Allow Attachments to be Deleted” on page 118
Initialize form at this Task	See “Initialize Form at this Task” on page 118
Customize Submit Action List	See “Customize Submit Action List” on page 118
Enable Form on Mobile Devices	See “Enable Form on Mobile devices” on page 119
Allow Viewing of Progress Map	See “Allow Viewing of Progress Map” on page 121
Display page for Entering Subject and Notes	See “Display Page for Entering Subject and Notes” on page 121
Automatically Display Next Form on Submit	See “Automatically Display Next Form on Submit” on page 121
Represent Static Table Cells as Top Level Fields	See “Represent Static Table Cells as Top Level Fields” on page 121

Configure a Form Copy Task

The Form Copy task is used to add a read-only copy of a form to a user's APA Inbox.

To configure a Form Copy task

1. Open the **Form Copy Properties** dialog box and click the **Form Copy** tab.
2. To select a form, click **Search**

   The **Search Forms** dialog box opens.
3. Browse to or search for the form you want to copy. Click the form, and then click **OK**. For information about how to search the APA server, see "Search the APA Server" on page 132.
4. Select or clear the following check boxes.
   - **Allow Viewing of Progress Map**. See "Allow Viewing of Progress Map" on page 121.
   - **Represent Static Table Cells as Top Level Fields**. See "Represent Static Table Cells as Top Level Fields" on page 121.

Configure a Form Export Task

The Form Export task is used to run all exports defined on the APA Server for a specific form.

To configure a Form Copy task

1. Open the **Form Export Properties** dialog box and click the **Form Export** tab.
2. To select a form, click **Search**

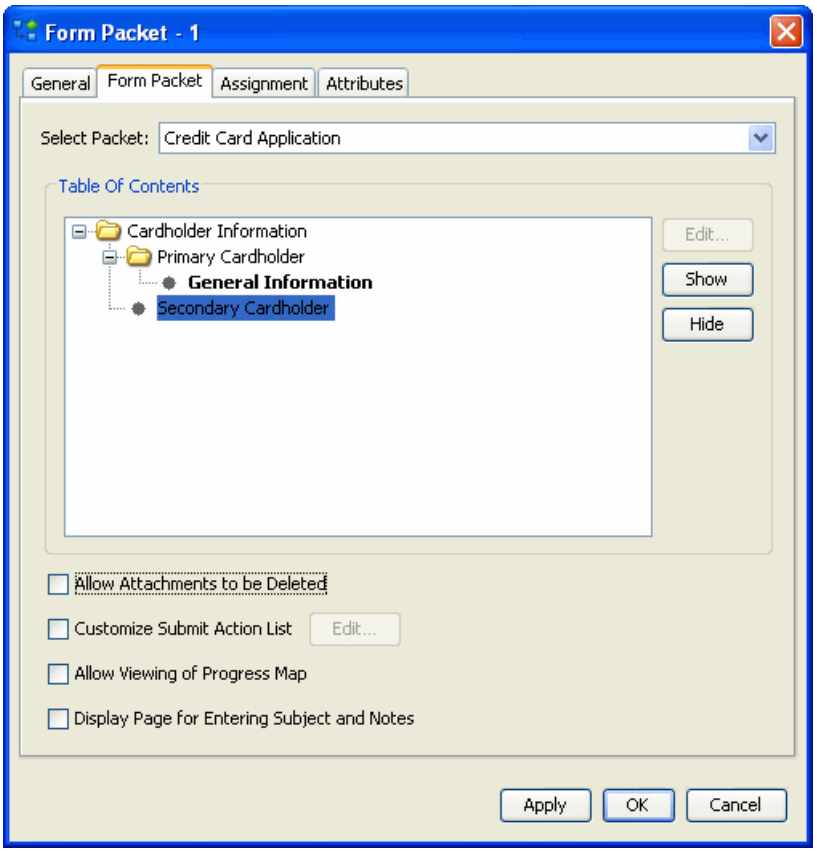
   The **Search Forms** dialog box opens.
3. Browse to or search for a form. Click the form, and then click **OK**. For information about how to search the APA server, see "Search the APA Server" on page 132.
4. Select or clear the following check boxes.
   - **Represent Static Table Cells as Top Level Fields**. See "Represent Static Table Cells as Top Level Fields" on page 121.

Configure a Form Packet Task

The Form Packet task is used to associate a Packet with a Task.

To configure a Form Packet task

1. Open the **Form Packet Properties** dialog box and click the **Form Packet** tab.



2. In the **Select Packet** list, click a Packet to associate with the task.
3. In the **Table of Contents** area, configure the form packet task.
   - To show a form, select a form in the table of contents and click **Show**.
   - To hide a form, select a form in the table of contents and click **Hide**.
   - To edit the properties of a form, use the following steps:
     i. Select a nodes with an associated form and click **Edit**.
     The **General Information** dialog box opens.
     ii. Configure the form options:

Allow Form to be Edited	See “Allow Form to be Edited” on page 117.
Initialize form at this Task	See “Initialize Form at this Task” on page 118
Set Field Properties	See “Set Field Properties” on page 117
Customize Submit Action List	See “Customize Submit Action List” on page 118
Enable Form on Mobile Devices	See “Enable Form on Mobile devices” on page 119
Automatically Display Next Form on Submit	See “Automatically Display Next Form on Submit” on page 121

iii. Click **OK**.

The **General Information** dialog box closes.

4. Configure the following form options (these options apply to all of the forms in the table of contents):

Allow Attachments to be Deleted	See “Allow Attachments to be Deleted” on page 118
Customize Submit Action List	See “Customize Submit Action List” on page 118
Allow Viewing of Progress Map	See “Allow Viewing of Progress Map” on page 121
Display page for Entering Subject and Notes	See “Display Page for Entering Subject and Notes” on page 121

Task Properties

The following sections describe some common task properties.

Use Business Calendar

The delay values for Delay and Escalate tasks can be entered in business time. If the delay time for a task is specified as 8 hours (using the business calendar), the path is followed after 8 business hours rather than 8 hours. This feature can be used to prevent the Delay or Escalate tasks being completed outside business hours, for example during the night or weekend.

The default business calendar calculates the time that has passed during business hours of 09:00 to 17:00, from Monday to Friday.

Java developers can define a custom business day and week using the `BusinessCalendarFactory.setDefaultBusinessCalendar( )` method found in the APA Process API. Refer to the *APA Programming Guide* for more information about this API.

Allow Form to be Edited

You can override default form settings by selecting the **Allow Form to be Edited** check box.

Multiple sources can edit form properties. The order of application is as follows, from highest priority to lowest:

- Process task settings
- Form server script
- Form client script

Set Field Properties

You can specify properties for form fields that override the default form settings.

- **Required**. The respondent must enter data in the field. An error message is displayed if the user attempts to Submit the form without completing the required fields.
- **Read-only**. The respondent cannot edit the field.
- **Hidden**. The user is unable to see the field on the form.

To set field properties

1. Select the **Set Field Properties** check box.
2. Click **Edit**.

   The **Edit Fields** dialog box opens. The list of fields in this dialog is built dynamically by the APA Server based on the selected form.

3. In the list, click a form field that you want to modify. You can select multiple fields by holding down the CTRL or SHIFT keys.
4. Select or clear the **Required**, **Read-only** or **Hidden** check boxes in the **Properties** area.
5. Click **OK**.

Allow Attachments to be Deleted

To allow users at the selected process step to delete attachments added by any user in the process, select the **Allow Attachments to be Deleted** check box.

Initialize Form at this Task

When a form is initialized, form fields are populated with initial values and profile data from the APA server. If the form uses any on-load database lookups, these are also executed.

In the **Initialize Form at this Task** list, click one of the options:

- **If first use** (Default). The form is initialized if it has not been previously opened, and then saved or submitted. The form does not initialize for the current task if it was opened and saved elsewhere in the process.
- **Always**. The form is initialized every time it is opened in this task, regardless of how this form is used in other tasks in this process.
- **Never**. If this option is selected, the form never initializes for this task.

Customize Submit Action List

Process Studio allows you to customize the Submit Action list of the forms associated with Form and Form Packet tasks. (For more information about submit actions, refer to the *Form Designer User Guide*).

To customize a Submit Action List

1. Click **Edit**.

   The **Submit Action** dialog box opens.

2. To add a custom submit action, click **Add**.

   A new submit action is added to the list.

   a. In the **Type** column, click a submit action type. (For information about default submit actions, see “Default Submit Actions” on page 119).

   b. In the **Display Name** column, type the text you want to display to the user. Press ENTER to save your changes.

c. In the **Value** column, type an output value for the submit action. Press ENTER to save your changes.

NOTE The value assigned to custom submit actions can be used with a Branch task to control which branch is selected, or with a loop task to control whether the loop continues.

3. To remove a custom submit action, select it and click **Delete**.
4. Click **OK**.

Default Submit Actions

Submit action	Description
Submit	Submits the form to the APA Server and, if applicable, the next user or group working with the form. The task is marked as complete and the next task in the process is executed.
Reject	The task is marked as complete and the next task in the process is executed.
Transfer	Transfers the form to another user or group. The task remains active.
Save	Saves the form to the user's Inbox on the APA Server. Incomplete forms are *not* moved to the Saved Items folder. If the form is incomplete it remains in the Inbox. The task remains active.
Cancel	Cancels the form submission. The task remains active.
Reset	Resets the fields on the form to the default settings.

Enable Form on Mobile devices

APA Mobile Option enables APA users to receive and submit in-process forms using mobile devices.

NOTE This feature is only available if you have the APA Mobile Option. Please see your Autonomy sales representative for additional information.

NOTE Ink Pictures and Buttons are not supported on mobile devices.

To display forms on mobile devices

1. Click one of the options in the **Enable Form on Mobile Devices** list:
   - **Yes**. The form is enabled for use on mobile devices. If you select this option, you must select the form fields that should be displayed. See "Configure the form fields to display on mobile devices" on page 120.
   - **No**. The form (when associated with this task) is not enabled for use with mobile devices.
   - **Use Form Defaults**. The form is enabled for use on mobile devices. The form's default settings determine the fields that are displayed on the mobile device. Click **Edit** to view these fields. The default settings can not be modified in Process Studio. To edit the default settings for the form, use the Management Console.

Configure the form fields to display on mobile devices

To configure how form fields are displayed on mobile devices, use the following procedure.

Set up the form fields to display on mobile devices

1. Click **Edit**.

   The **Mobile Properties** dialog box opens.

2. In the **Form Fields** list, select the fields that you want to display on mobile devices, and click .

   The selected form fields are moved to the **Mobile Display** list.

3. In the **Mobile Display** list, select any form fields that you do not want to display on mobile devices and click .

   The selected form fields are moved to the **Form Fields** list.

4. The form fields are displayed on the mobile device in the order they are displayed in the **Mobile Display** list. To reorganize the fields, use the move buttons:

   Move the selected fields to the top of the display order.

   Move the selected fields up the display order.

   Move the selected fields down the display order.

   Move the selected fields to the bottom of the display order.

5. Click **OK**.

Allow Viewing of Progress Map

To allow users at the selected process step to view the progress map in the APA Portal or APA Web Desktop, select the **Allow Viewing of Progress Map** check box. This option is available for Form, Form Copy, Message, and Manual tasks.

Display Page for Entering Subject and Notes

After a form is submitted, an HTML page can be displayed for the respondent to enter additional notes. To display this page, select the **Display Page for Entering Subject and Notes** check box.

Automatically Display Next Form on Submit

When a form is submitted, the APA server can automatically display the next form in the process. To automatically display the next form, select the **Automatically Display Next Form on Submit** check box. If there are no active successor Form tasks when this task completes, no additional forms are displayed.

Represent Static Table Cells as Top Level Fields

In earlier versions of APA, form fields located in static tables were stored as independent, top-level fields.

To enable support for processes that were designed in earlier versions of APA, select the **Represent Static Table Cells as Top Level Fields** check box. All form tasks imported from earlier versions of APA have this check box selected by default.

IMPORTANT This option changes the underlying mechanism used by the task to represent table data. When checked, each table cell is stored as an independent Field. The field name is generated using the following pattern: `DTB__<Table Name>_<row #>_<column name>`. Setting this option affects the current task only. Tasks with this option enabled do not share table data with tasks that do not have this option enabled.

IMPORTANT Static table fields saved with this option enabled are not available to tasks that do not also have this option enabled.

CHAPTER 8

Simulate a Process

Process Studio includes simulation tools that can be used to test the execution of a process before it is deployed to the APA server. This section describes Process Simulation.

- Flow Simulation
- Load Simulation

Flow Simulation

A flow simulation is used to test the execution of a process. A flow simulation shows what paths are taken through the process, and tests the links between tasks.

Create a Flow Simulation

To create a flow simulation in Process Studio, use the following procedure to create an instance of the process.

NOTE Deprecated script is not supported in flow simulation mode. Additionally, some API objects are not supported in flow simulation mode due to the fact that the complete server environment is not available.

To create a Flow Simulation

1. On the **Run** menu, click **Create Flow Simulation**. Alternatively, click **Create Flow Simulation**

   An instance of the template is created. The instance appears in the **Instances** tab.

   The state of each task is displayed by color-coding. See “Task states” on page 125. When an instance is created, all of the tasks are set to the Ready state.

2. On the **Run** menu, click **Activate Instance**. Alternatively, click **Activate Instance** . This starts the execution of the process instance.

3. After an instance is activated, tasks are set to the Done or Active states. Some tasks require manual input. See “Flow Simulation Commands” on page 125.

4. When the end task reaches the *Done* state, the simulation is complete.

Duplicate an Instance

Instances in Process Studio can be duplicated. This is useful if you want to test a different flow on the same process (for example, testing various Branch paths).

The **Duplicate Instance** command duplicates the current instance in flow simulation mode. Duplicating an instance puts the new instance in a Ready state. The instance is duplicated, but not executed.

To duplicate an instance

1. Click the instance in the **Instances** tab.
2. On the **Run** menu, click **Duplicate Instance**.

   Another version of the selected instance is created and appears in the **Instances** tab.

Delete an Instance

To delete an instance, use the following procedure.

To delete an instance

1. Click the instance in the **Instances** tab.
2. On the **Edit** menu, click **Delete**.

Task states

When a flow simulation is created, the state of each Task in the simulation is displayed using colored circles on the Task. Tasks can have the following states:

Task state	Description	Example
Ready	The task has not started.	
Active	The task has started but not completed.	
Done	The task has completed.	
Abort	The task has aborted due to an error (for example a form has been moved, disabled or deleted from the APA server, or a script error has occurred).	
Passive	The passive state applies to all tasks that are not required for completion of the process. This includes any task following a branch or multi-branch task that is not taken. This also includes any task preceding an OR task that has not completed after one of the other preceding tasks has completed.	

Task states in published processes

The state of tasks in a published process instance can be determined using the **Activities** > **Processes** command in the Management Console or by viewing the Progress Map in the Portal or Web Desktop.

Assigning and Reassigning Tasks

Process tasks can only be assigned to a user assigned the Process role or to the originator of the process. If a task is activated but has not been assigned or is assigned to an invalid user, the task is left in the active state and is marked as unassigned. The task is not aborted. Process administrators can use the **Activities** section of the Management Console to search for processes that have unassigned form tasks and reassign those tasks.

Flow Simulation Commands

In a flow simulation, manual input is required to change the state of some tasks.

To simulate the completion of a task, set the *Active* tasks to *Done*. Successor tasks are then allowed to execute.

To stop the process execution at the current point, change the state of the *Active* task to *Abort*.

To change the state of a task

- Select the task and click **Run** > **Set Done** or **Run** > **Abort**.
- Right-click the task and click **Set Done** or **Abort.**

Change the State of Aborted Tasks

To change the state of an aborted task and continue the simulation, use the following commands. An aborted task can be forced to the *Ready*, *Active*, *Done*, or *Passive* state.

To change the state of an aborted task

- Select the task and click **Run** > **Force State**. Then click one of the task states.
- Right-click the task and click **Force State**. Then click one of the task states.

Cancel a Task

To cancel a task and change its state to *Passive*, use the following commands.

To cancel a task

- Select the task and click **Run** > **Cancel**.
- Right-click the task and click **Cancel**.

Open a Gate Task

The Gate task prevents execution of its successor tasks based on an arbitrary condition. When the gate is opened, the successor tasks can execute.

Gate tasks in process instances on the APA server are opened programmatically. When the gate is open, its successor tasks can execute.

To open a gate task in a flow simulation

- Right click the gate task and click **Open Gate**.

Load Simulation

A load simulation is used during the process design phase to analyze how a process might respond when deployed under load. The simulation is used to identify any potential bottlenecks early in the process.

A load simulation is also a useful tool when changes to a process are being contemplated. The simulation can be used to analyze the impact of changes before they are made.

One example of an issue identified through load simulation is excessive loading of specific resources, which could later manifest as a bottleneck caused by excessive queue times.

Configure Tasks for Load Simulation

Before creating a load simulation, the tasks in the process must be configured. Simulation settings are located in the **General** and **Simulation** tabs of the task **Properties** dialog box. The tasks that need to be configured are:

- Branch
- Command
- Committee
- E-mail
- Form
- Form Copy
- Loop
- Manual
- Message
- Script
- Sub-process
- Web Service

To configure the simulation properties for loop or branch tasks for, see the following topics:

- “Configure Simulation Properties for Loop Tasks” on page 128.
- “Configure Simulation Properties for Branch Tasks” on page 128

To configure the simulation properties for other tasks, use the following procedure.

To configure simulation properties

1. Double-click the task.

   The task **Properties** dialog box opens.
2. Click the **General** tab.
3. In the **Duration** box, type or select an estimate for the length of time required to complete the task.

IMPORTANT The resources defined in a load simulation are assigned to the entire duration of each task. For example, you have a process with one resource (User_1) and two tasks (Task_A with a duration of 24 hours and Task_B with a duration of 5 minutes). If Task_A is selected to be completed first, 24 hours are required before User_1 is released to complete Task_B.

4. Click the **Simulation** tab.
5. Move the **Standard Deviation** slider to change the Standard Deviation. Standard deviation is a statistic used to measure the variation in a distribution. The standard deviation simulates the fact that sometimes tasks are completed more quickly or slowly than the estimate.

   The changes you make using the slider are reflected in the **Standard Deviation** box (where you can also type or select a value) and on the chart.
6. Click **OK**.

Related Topics

- “Configure Simulation Properties for Loop Tasks” on page 128
- “Configure Simulation Properties for Branch Tasks” on page 128

Configure Simulation Properties for Loop Tasks

To configure the simulation properties for loop tasks, use the following procedure.

To configure simulation properties for Loop Tasks

1. Double-click the Loop Task to open the properties dialog box.
2. Click the **Simulation** tab.
3. Define the probability of the loop being executed by typing or selecting a value in the box. The value must be a percentage (between 1-100).
4. Click **OK**.

Related Topics

- “Configure Tasks for Load Simulation” on page 127

Configure Simulation Properties for Branch Tasks

To configure the simulation properties for branch tasks, use the following procedure.

To configure simulation properties for Branch Tasks

1. Double-click the Branch Task to open the properties dialog box.
2. Click the **Simulation** tab.
3. To define the probability of each Branch Path being taken, double-click each cell in the **Probability** column and type a percentage. The total of the values entered in the Probability column must equal 100%.
4. Click **OK**.

IMPORTANT If the total of the Probability values do not equal 100%, the value of the last Branch path is adjusted to make the total 100%

Related Topics

- “Configure Tasks for Load Simulation” on page 127

Create and Run a Load Simulation

Before creating a load simulation, verify that each task has been configured for simulation. See “Configure Tasks for Load Simulation” on page 127.

NOTE Sub-process tasks are ignored by load simulation. The **Invoke in Simulation** feature for sub-processes is supported only in flow simulations, not load simulations.

To create a load simulation

1. On the **Run** menu, click **Create Load Simulation**.

   The **Configure Simulation** dialog box opens.
2. In the **Delay Between Instantiations** area, type or select a **Mean** (average) value. The **Delay Between Instantiations** simulates how often a process is used. Reducing the **Mean** value simulates a process that is used more frequently (it is placed under greater load). The default values simulate a mean delay of 50 seconds between each instance, with a standard deviation of 10 seconds.
3. In the **Standard Deviation** box, type or select a standard deviation for the delay between instantiations.

NOTE Standard Deviation is a statistic used to measure the variation in a distribution.

4. Type or select the **Number of Instances** you want to run during the load simulation.
5. Move the **Time Compression** slider to increase or decrease the speed of the simulation. The maximum time compression runs the simulation in the shortest time. The minimum time compression results in a longer, but possibly more accurate, simulation.

   When you move the slider, the **estimated simulation duration** displays the estimated time to complete the simulation.
6. The **Configure Roles** area allows you to specify the number of users.

   To add users to the simulation, click **Add**.

   The **Configure Roles** dialog box opens.

   a. In the **Number of Users** box, type or select the number of users.

   b. Select the check boxes for the type of users you want to add to the simulation. All Roles, Relationships and Work Queues assigned within the process are available.

   c. Click **OK**.

   The Roles you have configured are displayed in the **Configure Roles** list. You might need to repeat these steps to add further types of user to the simulation.
7. To delete a configured role, click the role and click **Delete**.
8. To edit a configured role, click the role and click **Edit**.
9. To run the load simulation, click **Run**.

   The **Simulation** dialog box opens. Depending on the Time Compression used, the simulation may take a few minutes to complete.
10. To save the data generated by the simulation, click **Save Data**. This allows you to save the data in a comma delimited file (**.** CSV) file on your hard drive or a network location.
11. Click **Close**.

CHAPTER 9

Connect to an APA Server

This section describes how to connect to the APA Server.

- Connect to the APA Server
- Disconnect from the APA Server
- Search the APA Server

Connect to the APA Server

A connection to the APA Server is necessary to publish a process or search the APA Server.

To connect to an APA Server

1. Click the **Connect** command:
   - On the **Tools** menu, click **Connect**.
   - Click **Connect** .

   The **Login** dialog box opens.
2. In the **Server** box, type or select the server name.
3. In the **User Name** box, type your user name.
4. In the **Password** box, type your password.

5. If you would like your system to remember your password, select the **Remember my password** check box.
6. Click **OK**.

   A connection is established to the APA Server.

Related Topics

- “Search the APA Server” on page 132
- “Disconnect from the APA Server” on page 132

Disconnect from the APA Server

To disconnect from the APA Server, use one of the following commands.

To disconnect from the APA Server

- On the **Tools** menu, click **Disconnect**.
- Click **Disconnect** .

Related Topics

- “Connect to the APA Server” on page 131

Search the APA Server

When configuring tasks or publishing a process you might need to search the APA server. The Search feature enables you search for:

- published forms and processes.
- APA users.
- groups (for APA servers integrated with LDAP).
- work queues.
- relationships.

NOTE You must be logged on to the APA server before you can search. See "Connect to the APA Server" on page 131.

NOTE The folders and files that you can search are restricted by the permissions configured by your APA Administrator.

TIP If you are searching the APA server for a published form or process, you can also browse the folders on the server. See "Browse Folders on the APA Server" on page 134.

To search the APA server, use the following procedure.

To search the APA server

1. In the **Search for** box, type a search term, for example **`Expense_Report.`**

NOTE You can use the asterisk (*) character as a wildcard. For example, type * to search for everything.

NOTE You must type at least one alpha-numeric character or the wild-card character * in the **Search for** box to perform a search.

2. In the **in** list, click an option to define where to search. (The options that are available depend on the type of search).

Everywhere	All of the following areas.
Process	Search Process names.
Name	Search Form names.
Folder	Folder names.
Author	Form or Process author names.
Publisher	Form or Process publisher names.
Keywords	Form or Process keywords.
Description	Form or Process descriptions.
Form Text	All form text (for searching forms).
UserID	User IDs (for searching Users).

First Name	First Names (for searching Users).
Last Name	Last Names (for searching users).
Last Name, First Name	Last Name, First Names (for searching users).
Work Queue	Work Queues.

3. If you are searching for Users, you can also specify further options.

Contains	Search for words that contain the search term.
Starts With	Search for words that start with the search term.
Exact Match	Search for an exact match of the search term.

4. Click **Search**.

   The search results are displayed in the list. If necessary you can search again.

5. Select one or more of the search results:
   - To select one of the search results, click the item and click **OK**.
   - In some circumstances you are able to select multiple items.
     i. Select one or more items in the list and click **Add**.

        The items are added to the **Selected Items** list. If required, you can search the APA server again. To remove an item from the **Selected Items** list, click the item and click **Remove**.

     ii. Click **OK** to confirm your selection.

Browse Folders on the APA Server

When the **Search** dialog box is used to select forms or processes, you can browse the folders on the APA server.

- To open a folder, double-click its name in the list.
- To move up a level in the directory structure, click .

CHAPTER 10

Publish a Process

This section describes how to publish a process to the APA server.

- Publishing Overview
- Publish a Process
- Publishing Wizard: General Properties
- Publishing Wizard: Permissions
- Publishing Wizard: Process Settings
- Initiate a Process

Publishing Overview

After a process has been created and tested, it can be published to the APA server so that it can be used.

The following conditions must be met before a process can be published:

- You must be logged on to the APA server. See “Connect to the APA Server” on page 131.
- Only APA users assigned the *Process Publisher* role are permitted to publish processes. Refer to the *Management Console User Guide* for more information about Roles.

- You must have the relevant permissions for the folder on the APA server where you want to publish the process.
- The process must be saved before it can be published. See “Save a Process” on page 48.

Publishing and Permissions

The Publishing Wizard displays all the folders on the APA server. However, users cannot see forms or processes in folders for which they do not have publishing rights.

If you select a folder to which you do not have publishing rights, the controls on the **Publishing Wizard** dialog box are disabled. A lock icon on the selected folder indicates the lack of publishing rights.

You can add, delete, or rename folders on the APA server. You can delete empty folders to which you have publishing rights. You can not delete or rename folders that contain forms or processes.

When a new sub-folder is created, it inherits publishing permissions from its parent folder.

Publish to Multiple Servers

APA Process Studio stores references to process forms by folder and form ID. This allows process definitions to portable between APA Servers. A single process definition can be published to multiple servers and used successfully if each server has common folder names and form IDs, as used in the process definition.

The need to publish process definitions to multiple servers may arise if you use a test server prior to deploying process definitions to a production server. Form references are converted from folder and form ID to form GUID on a specific server when a process definition is published.

APA Process Studio stores references to work queues by name rather than by the internal numeric ID used by each individual server. A single process definition can be published to multiple servers and used successfully if each server has common work queue names, as used in the process definition. Work queue references are converted from name to internal numeric ID on a specific server when a process definition is published.

Publish a Process

To publish a process, use the following procedure.

To publish a process to the APA Server

1. Connect to the APA server. See “Connect to the APA Server” on page 131.
2. On the **File** menu, click **Publish Process**.

   If any of your tasks are missing required fields, a dialog box opens describing the problem. The publishing process is aborted. Click **OK**, fix the fields on the applicable tasks, and then try to publish again.

   The **Publishing Wizard** dialog box opens and the **General Properties** page is displayed.

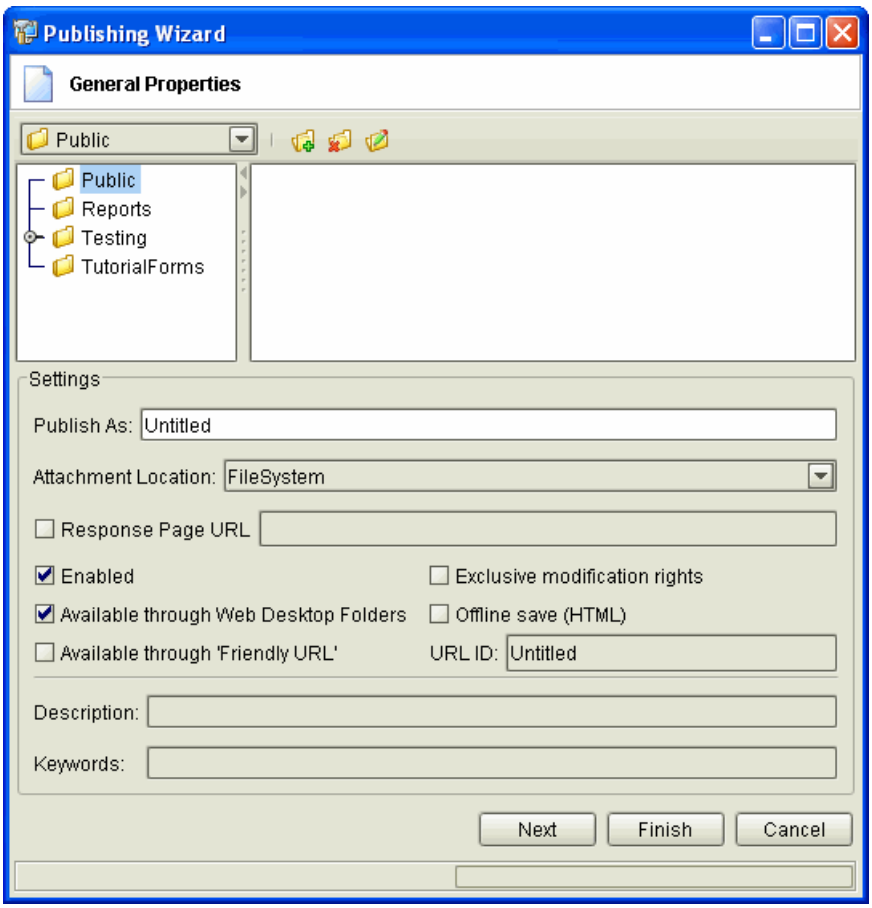
   Continue with “Publishing Wizard: General Properties” on page 138.

Related Topics

- “Publish to Multiple Servers” on page 136.
- “Publishing Wizard: General Properties” on page 138.

Publishing Wizard: General Properties

The first page of the **Publishing Wizard** is used to configure the General Properties of the process.



To configure the General Properties for the published process

1. Navigate to the location on the APA server where you want to save the published process.

   To create a new folder, delete a folder, or rename a folder, use the toolbar buttons:

Icon	Description
	Create a subfolder in the selected folder
	Delete the selected folder
	Rename the selected folder

2. In the **Publish As** box, type a name for the published process.
   - If you use the same name as an existing process in the same folder, that process is overwritten.
   - The process name can not exceed 128 characters.

IMPORTANT Processes can be published over existing forms or processes.

When a process is published over a form, all permissions, routing settings, and export settings are removed. This ensures that if another form is published over the process, the new form does not inherit these settings from the original form.

3. In the **Attachment Location** list, click a location to define where attachments are stored.
   - If you select **FileSystem** the attachments are stored in *`Server Resources/`*`attachments`. The location of the server resources directory is specified when APA Server is installed.
   - If you are exporting data to a Document Management System, click a Document Management System Connect Agent.

     If you select a Document Management System Connect Agent, files attached to forms are uploaded to the Document Management System in the directory provided when the Document Management System Connect Agent was configured (with the exception of URL attachments, which are stored in the APA ROUTEATTACH database table). If you select a Document Management System that does not allow the user to be automatically authenticated using the APA Form, they will need to set up their Document Management System User Name and Password. (For more information, refer to the *APA Portal online help* or *Web Desktop online help*).
4. **Response page URL**. After a process is initiated, a response page is displayed. To use a custom response page, select the **Response page URL**

check box and type the URL for the response page you want to use. The URL can not exceed 250 characters.

NOTE If a process is initiated and an initial form task is not specified in the process definition, the user is redirected to the response page URL (or the default response page). If an initial form task is specified in the process definition, the form is displayed and the response page URL is ignored.

5. **Enabled**. To enable the process (so that it can be used by the users configured on the **Permissions** page) select the **Enabled** check box.
6. **Available through Web Desktop Folders**. To allow the process to be accessed through the Web Desktop or Portal, select the **Available through Web Desktop Folders** check box.
7. **Available through 'Friendly URL'**. To specify a user-friendly URL for the process, select the **Available through 'Friendly URL'** check box. Type a URL in the **URL ID** box.

   The URL must contain only legal URL characters. Spaces are converted to underscores. The URL ID must be unique throughout the entire APA system (not just in the selected folder). The URL can not exceed 250 characters.

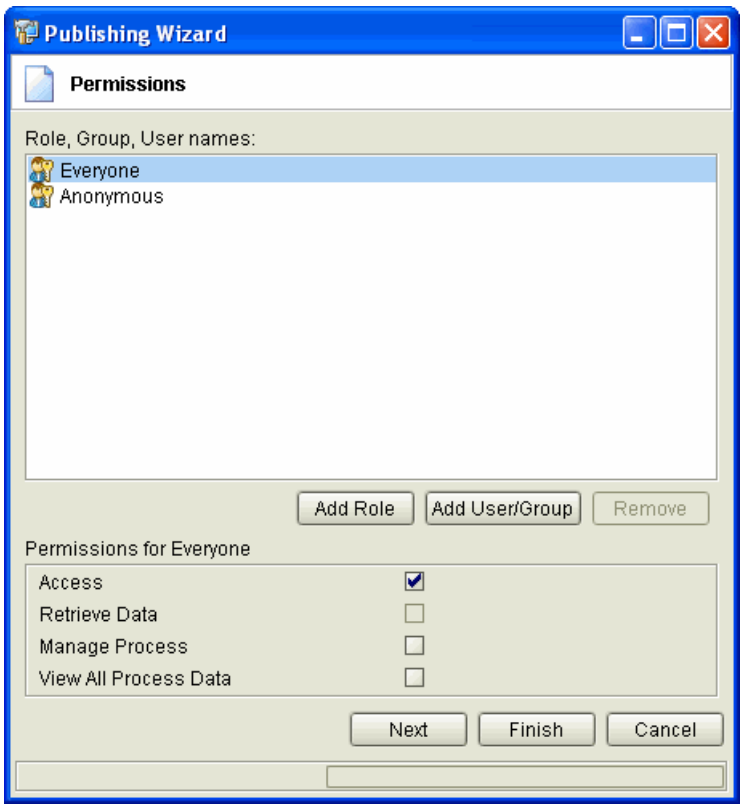
NOTE If the URL ID is not unique then the APA Server appends characters to make it unique.

8. **Exclusive modification rights**. To prevent anyone besides the publisher (you) and APA administrators from modifying the process, select the **Exclusive modification rights** check box.
9. The **Description** and **Keywords** assigned to the process are displayed. This information is read-only. To edit the description and keywords, see "General Process Properties" on page 52.
10. Click **Next**.

    The **Permissions** page is displayed.
11. Continue with "Publishing Wizard: Permissions" on page 141.

Publishing Wizard: Permissions

Permissions are used to control how APA users can interact with published processes. Permissions are defined on the **Permissions** page of the **Publishing Wizard** when the process is published.



After a process has been published, its permissions can be modified using the Management Console.

Configure Permissions

To configure the permissions of a process, use the following procedure.

To modify permissions

1. Click a user, group, or role in the **Role, Group, User names** list. For information about how to add users to this list, see “Roles, Groups, and Users” on page 142.

2. In the **Permissions** area, select the check boxes for the permissions that you want to grant to that user. Clear a check box to remove a permission. For more information about permissions, see “Permissions” on page 143.

NOTE If a permission is available to a user or group by default, the check box is selected and unavailable.

3. After you have configured permissions:
   - To continue the publishing process, click **Next**. The **Process** page of the **Publishing Wizard** dialog box opens. Continue with “Publishing Wizard: Process Settings” on page 145.
   - Click **Finish** to accept the default Process options and publish the process to the APA server.
   - To stop the publishing process, click **Cancel**.

Roles, Groups, and Users

The **Role, Group, User names** list shows the roles, groups and users whose permissions can be configured. It includes the Everyone and Anonymous roles by default.

- **Everyone**. Any user who logs on to the APA Server.
- **Anonymous**. Anyone who browses to the APA Server and does not log on.

NOTE You should test your process before granting **Anonymous** or **Everyone** access permission.

Additional users can be added to the list.

Add a user or LDAP group

To add a user or group to the **Role, Group, User names** list, use the following procedure.

To add a User or Group

1. Click **Add User/Group**.

   The **Search Users** dialog box opens.

2. Search for or select any user or group. See “Search the APA Server” on page 132.
3. Click **Add**.

   The user or group is displayed in the **Selected Items** list. If required, you can continue to add further users and groups.
4. Click **OK**.

   The selected users are added to the **Role, Group, User names** list on the **Permissions** page.

Add a Role

To add a role to the **Role, Group, User names** list, use the following procedure.

To add a Role

1. Click **Add Role**.

   The **Search Roles** dialog box opens.
2. Click a role in the list, and click **Add**.

   The role is displayed in the **Selected Items** list. You can add further roles if required.
3. Click **OK**.

   The selected roles are added to the **Permissions** page.

Permissions

The **Permissions** area shows the permissions for the role, user or group selected in the **Role, Group, User names** list and is used to change the permissions assigned to those users.

Users can be assigned a combination of the following permissions.

Access	Allows users to open a form or initiate a process. By default, the *Everyone* role is assigned this permission. If the Access permission is granted to *Anonymous* users, all other users, groups and roles inherit this permission.
Retrieve Data	Allows users to retrieve data through the My Data Client. (Refer to the *My Data Client User Guide* for more information). To retrieve data through My Data Client, a user must also be assigned the *My Data* role. By default, the *Process Publisher* is assigned this permission.
Manage Process	Allows users to manage the process in the **Activities** section of the *Management Console*. To manage the process, a user must also be assigned the *Process Administrator* role. By default, the *Everyone* role is assigned this permission.
View All Process Data	Allows users to open a form associated with a process and view the current form data. To make use of this permission, a user must also be assigned the *Process Administrator* role. By default, the *Everyone* role is assigned this permission.

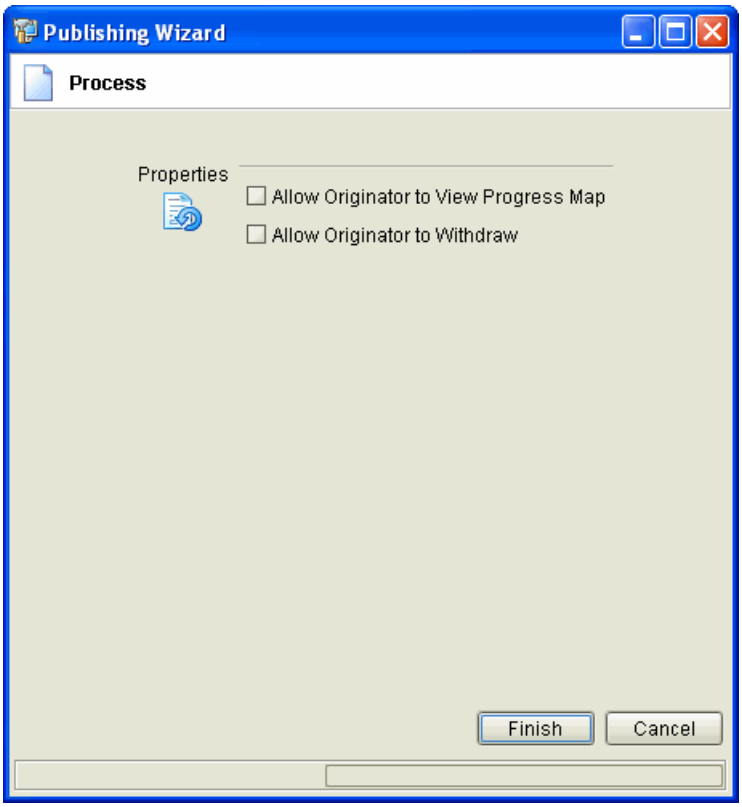
NOTE The only permission allowable for LDAP groups and *Anonymous* is Access. The remaining permissions require a user ID on the APA Server.

Precedence of Form Tasks and Process Permissions

The **Manage Process** and **View All Process Data** permissions can apply to either a form task or a process. If a form task is associated with a process, both (**Manage Process** and **View All Process Data**) permissions set on the process take precedence over the permissions set on the form.

Publishing Wizard: Process Settings

The Publishing Wizard displays the **Process** page.



To configure Process settings

1. To allow the process originator to view the progress map for the process, select the **Allow Originator to View Progress Map** check box. The progress map is viewed from the Sent Items folder in the Portal or Web Desktop. The progress map shows a live picture indicating the current state of the process and tasks.
2. To allow the process originator to withdraw a form, select the **Allow Originator to Withdraw** check box. Forms can be withdrawn from the Sent Items folder in the Portal or Web Desktop. Withdrawing a form removes it from the approval process.
3. Click **Finish**.

   A dialog box opens confirming that the process has been published to the APA server.

4. Click **Open** to launch the process, or click **Close** to close the Publishing Wizard.

Initiate a Process

Initiating a process creates an instance of a process template on the APA Server. You can not initiate a process until the process has been published to the APA Server. Processes coexist with published forms and external documents in the Folders repository of the APA Server. You can access these published processes using the APA Portal or Web Desktop.

To initiate a Process from APA Portal

1. Log on to the Portal. Refer to the *APA Portal User Guide* for more information.
2. In the **Navigation** area, click **Folders**.
3. Navigate to the process you want to initiate and click the process name.

   The applicable form is displayed or the following message is displayed: "`A new instance of process <process_name> has been initiated`."

   Complete the form or click **Close Window**.

Initiate a Process from the Web Desktop

To initiate a process using the Web Desktop, use the following procedure.

To initiate a Process from the Web Desktop

1. Log on to the Web Desktop. Refer to the *Web Desktop User Guide* for more information.
2. Click the **Folders** tab.
3. Navigate to the process you want to initiate and click the process name.

   The applicable form is displayed or the following message is displayed: "`A new instance of process <process_name> has been initiated`."

   Complete the form or click **Close Window**.

Other Methods to Initiate a Process

You can also use the following methods to initiate a process:

- **URL link**. A URL link (for example, in an e-mail message) can be used. Click the link to initiate the process. If a process is initiated by a URL link, additional parameters can be specified in the URL to initialize the corresponding fields in the new process.

  A sample URL link might look like this: `http://myserver/lfserver/myFriendlyURL?field_name1=value1&field_name2=value2.`

- **Virtual Submit**. Virtual Submit can be used to initiate a process. The 'DFS__FormGUID' parameter must refer to a process definition GUID as generated at publish time. The values of data fields in the Virtual Submit XML document are used to preset values for process attributes. Virtual Submit is only used to initiate a process. It does not complete any tasks in the process. Refer to the *Management Console online help* for additional information about Virtual Submit.

- **SOAP**. The SOAP call '`getFormData`' can also be used to initiate a process. The '`formGUID`' parameter must refer to a process definition GUID as generated at publish time. The '`xmlData`' parameter may optionally be used to preset values for attributes. Refer to the *APA Programming Guide* for more information.

- **POST**. When using POST to initiate a process:

  - The URL posted to must be a valid URL for requesting a form (for example, Friendly URL).
  - Field values supplied in the POST operation are used to prepopulate form fields (they are used as initial values).
  - Field values supplied as POST could conflict with field values supplied on the URL. You should avoid supplying the same field using both POST data and URL parameters because the results usually appear on the server as multiple values for the repeated field ID. Which one gets used first for a single-value field is unpredictable.
  - Field values supplied on the URL are not used at all if multi-part POST is used to upload files.
  - Fields supplied using POST or the URL must have a corresponding field defined in the form. Otherwise the data for these fields is discarded.
  - If multiple values are provided for a field but the field does not support multiple values (for example, single select list box), the results are unpredictable.
  - Field values must conform to the type expected by the form field. Otherwise the data may be lost or not formatted correctly.

- Most system fields (including `LF__UserID` and `DFS__FormGUID`) are set by the server and any values supplied via POST or URL might be overridden by the server value.

Order of Initialization

When a process is initiated, operations take place in the following order:

- Any values from virtual submit or form POST operation are applied.
- Profile field values are applied to any relevant fields that are still blank.
- Static initial values from the form definition are applied to any fields that are still blank.
- Any remaining initializations (including lookups and script) are applied normally.
- Process field property settings (read only, required, and hidden) are applied. These are set in Form Designer and can be edited in Process Studio.

PART 3 Process Studio Reference

This section provides reference information for Process Studio.

- Task Patterns
- APA Expression Language

CHAPTER 11

Task Patterns

Task Patterns are groups of tasks that can be used to accomplish a specific function in a process.

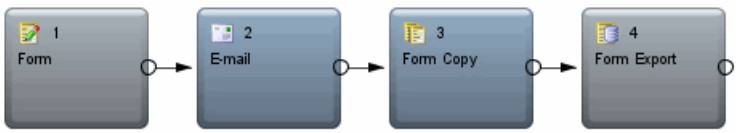
- Simple Design Patterns
- Task Patterns

Simple Design Patterns

This section demonstrates some simple design patterns.

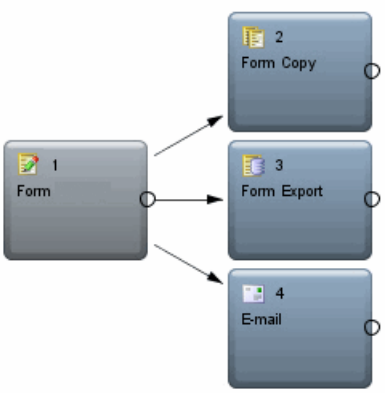
Sequence

When tasks are connected in sequence, a task must complete before its successor task can start.

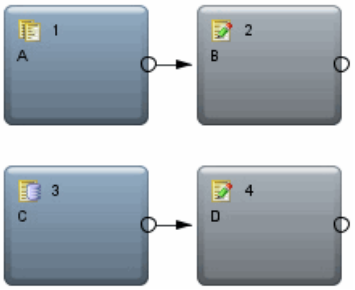


Parallelism

Sometimes several tasks can be active, in parallel, at the same time. This occurs when a task has more than one successor. In the following example, when the Form task completes, the Form Copy, Form Export, and E-mail tasks are executed in parallel.

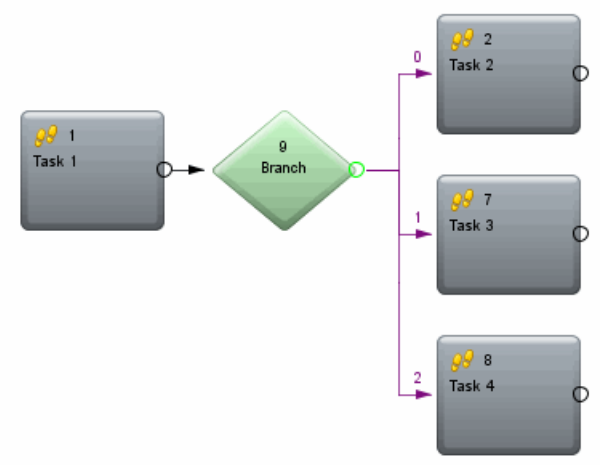


Tasks can also execute in parallel without a split. If you have multiple tasks with no predecessor (more than one start task), the two paths are automatically executed in parallel. In the following example, tasks A and C execute in parallel.



Branch

A process can split into more than one path. A branch task is used to decide which path is taken. A branch task can be configured to select a path based on a default value, an expression, or the result of another task.

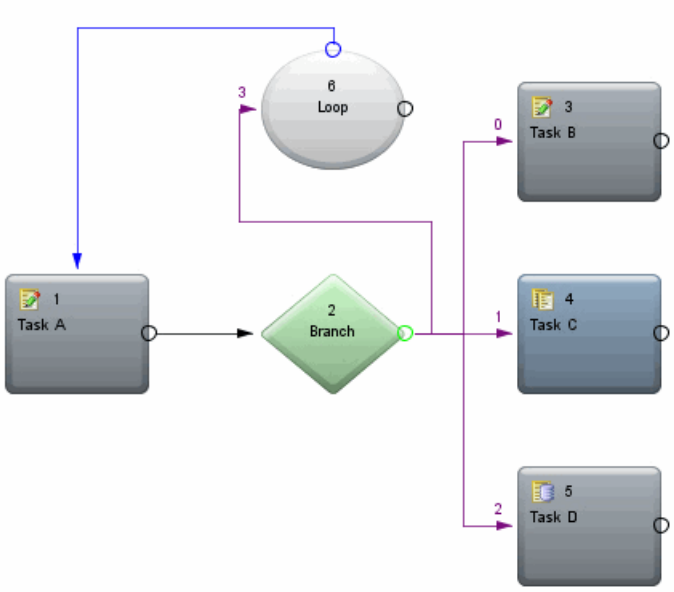


Related Topics

- “Branch Task” on page 74.

Loop

A loop allows part of a process to be used repeatedly. Loops are created using a loop task.



Related Topics

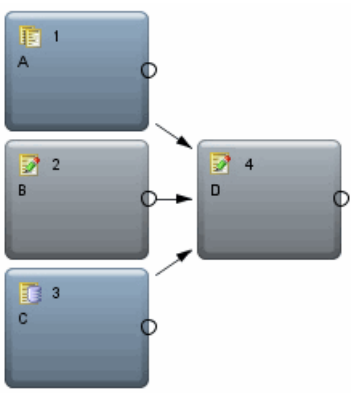
- “Loop Task” on page 74.

Join

When two or more tasks must be complete before another task can begin, it is necessary to create a join. Tasks can be joined using the And task or connected directly to a successor.

If you do not use an And task to join two or more predecessor tasks to a single successor task, the presence of an And task is implied (it is default behavior that a task does not start until all of its predecessors are complete).

In the following example, tasks A, B, and C must complete before Task D can begin.



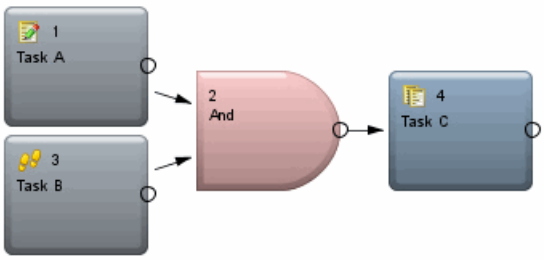
Task D is executed only one time. It does not execute for each predecessor task.

Related Topics

- “And Task” on page 72.

And

The And task requires its predecessor tasks to be complete before the process can proceed. In the following example, tasks A and B must complete before task C can begin.

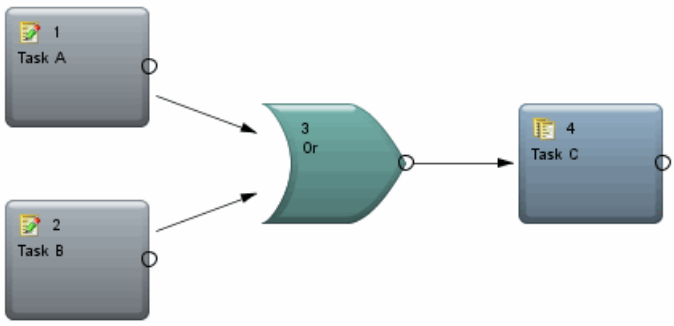


Related Topics

- “And Task” on page 72.
- “Join” on page 154

Or

The Or pattern requires only one of its predecessor tasks to complete before its successor task can start. In the following example, as soon as either task A or B is complete, task C begins.



This pattern is created using an OR task.

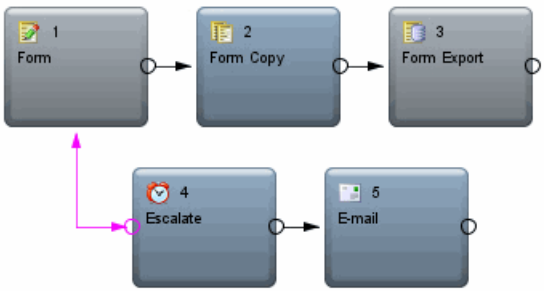
Related Topics

- “Or Task” on page 73.

Escalate

The Escalate pattern is used to ensure that important tasks are completed. The Escalate task provides a secondary process path that is used if a task is not completed within a fixed amount of time.

In the following example, if the Form task does not complete within the delay time specified in the Escalate task, the Escalate path is used. The E-Mail task becomes active and an e-mail is sent to remind the relevant user to complete the form.



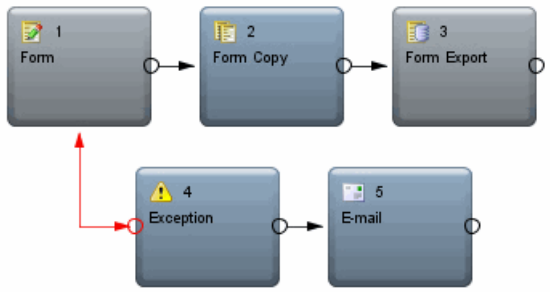
Related Topics

- “Escalate Task” on page 75.

Exception

The Exception pattern is used to handle errors in a process. The Exception pattern provides a secondary path that is used if a task enters the *Aborted* state.

In the following example, if the Form task fails, the Exception path is used. The E-Mail task becomes active and an e-mail is sent (for example, to an administrator).



The Exception pattern is useful for implementing script error handling.

Related Topics

- “Exception Task” on page 76.

Task Patterns

Process Studio includes many advanced Task Patterns in the Patterns drawer of the Task Library.

To add a pattern to your process template, see “Add a Task Pattern” on page 82.

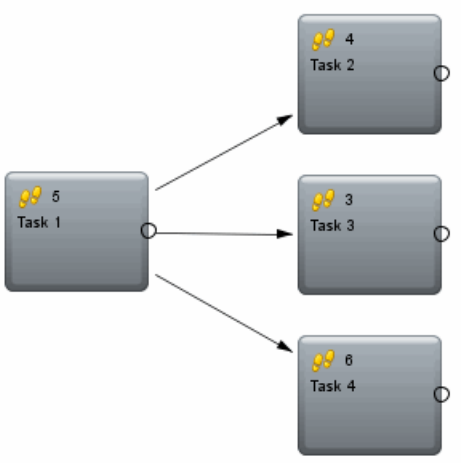
The patterns described in this section are based on those by W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros, *Workflow Patterns*, http://www.workflowpatterns.com.

Sequence

In a sequence, a task must be completed before the following task becomes active.

Parallel Split

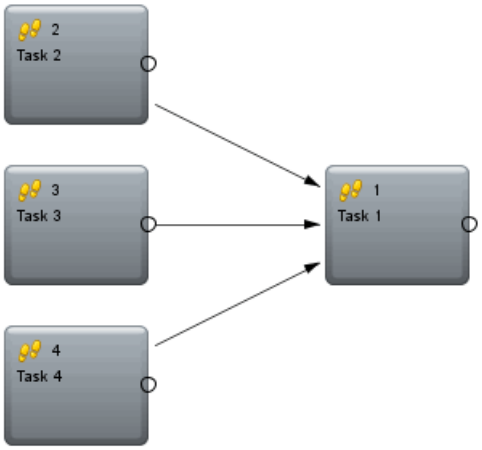
Tasks can be active, in parallel, at the same time. This occurs when several tasks are initiated on completion of one task. In the following example, when task 1 is complete, tasks 2, 3, and 4 execute in parallel.



Synchronization

Synchronisation occurs when two or more tasks must be complete before another task can begin. Synchronisation can be accomplished using an And task.

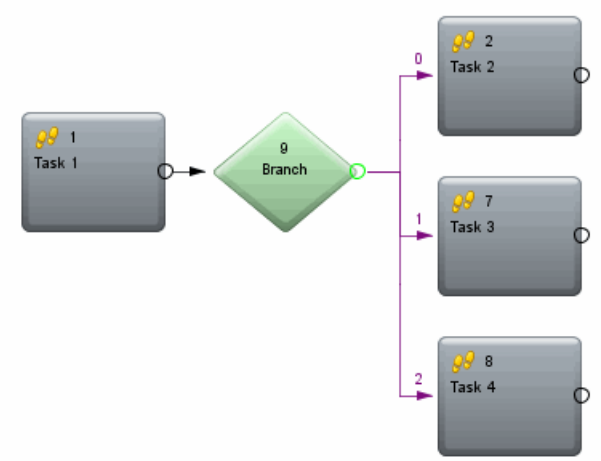
If you join multiple predecessor tasks to a single successor task (as in the following diagram) an And task is implied. All of the predecessor tasks (2, 3, and 4) must complete before the successor task (1) can start. The successor task is only executed one time.



If you want to connect multiple tasks to a single successor task, and execute the successor when the first of the tasks is complete, use an OR task. See “Or Task” on page 73.

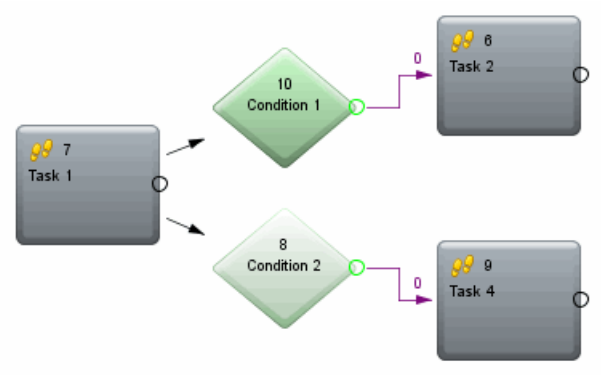
Exclusive Choice

An exclusive choice occurs when only one of several paths through a process is followed. An exclusive choice is created using a Branch task.



Multi-choice

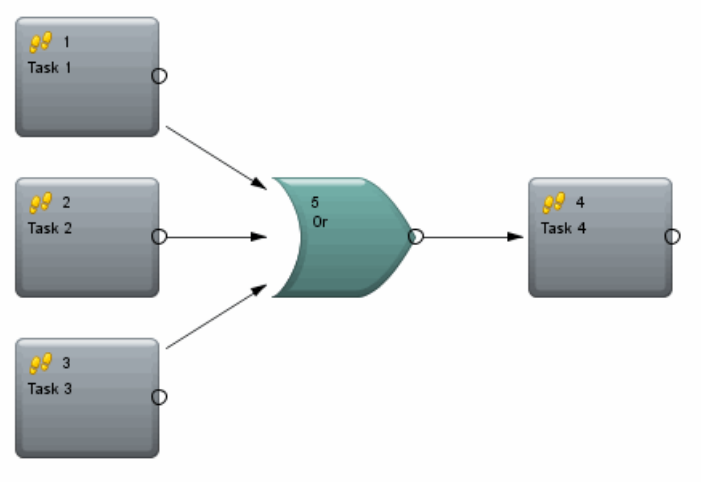
The multi-choice pattern allows you to create a situation where more than one process path is followed, if certain conditions are met. Branch tasks are used to control whether each path becomes active.



Simple Merge

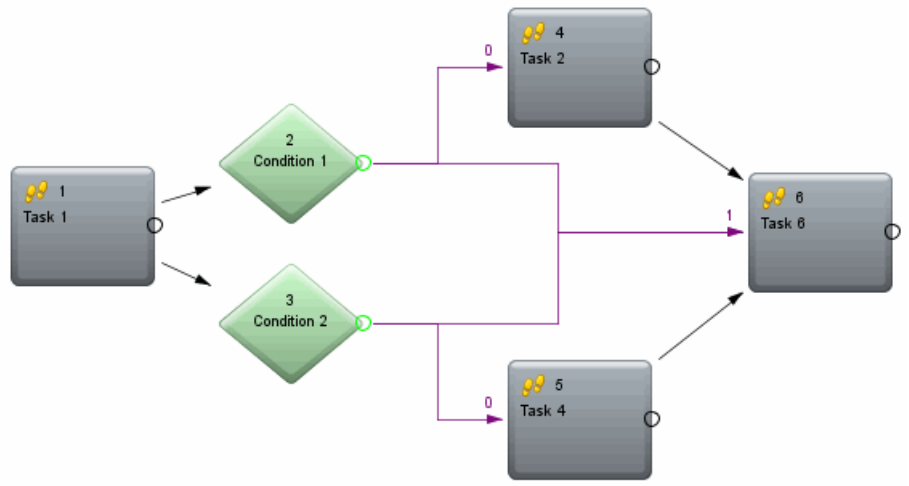
A simple merge occurs when two or more paths in a process meet. Once one of the paths reaches the merge point, the others are ignored.

A simple merge is modeled using an Or task (with **Cancel unfinished predecessors as soon as any predecessor is done** selected), as shown below. As soon as Task 1, 2, or 3 completes, Task 4 becomes active and the remaining tasks become passive.



Synchronizing Merge

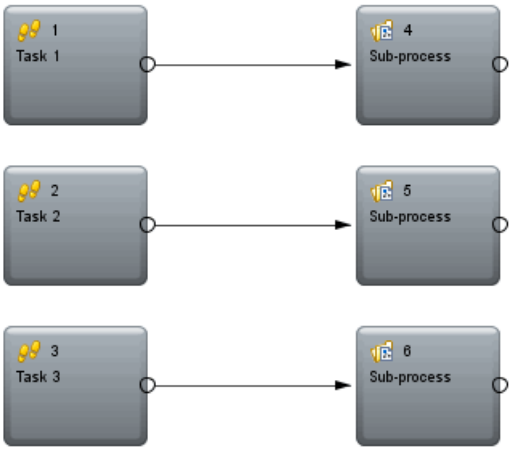
The Synchronizing merge pattern combines a multi-choice pattern with the synchronization pattern. In the example, Task 2 and Task 4 may or may not execute. If one task executes, Task 6 does not become active until it has completed. If both execute, Task 6 does not become active until they have both completed.



Multi-merge

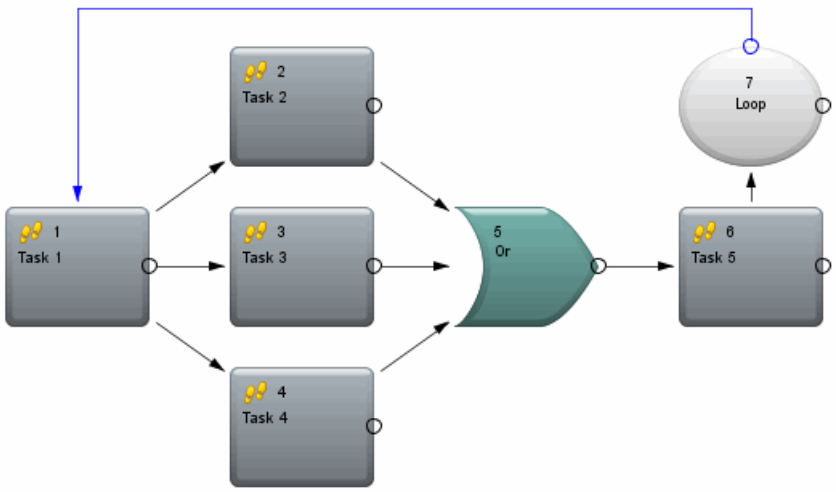
A multi-merge occurs when process paths converge. Normally, a task is not executed repeatedly as each of its predecessors completes. Instead it executes a single time once all of its predecessors are complete (see “Synchronization” on page 158). In a multi-merge the successor task *is* executed once for each predecessor.

A multi-merge can be implemented using a Sub-process (with **Embed Sub-process Instance in Main Process** selected) for the activity following the merge, with a Sub-process task for each merging activity. Each of the three Sub-process tasks has a target of the same process. As each Task (1, 2, and 3) completes, the Sub-process executes.



Discriminator

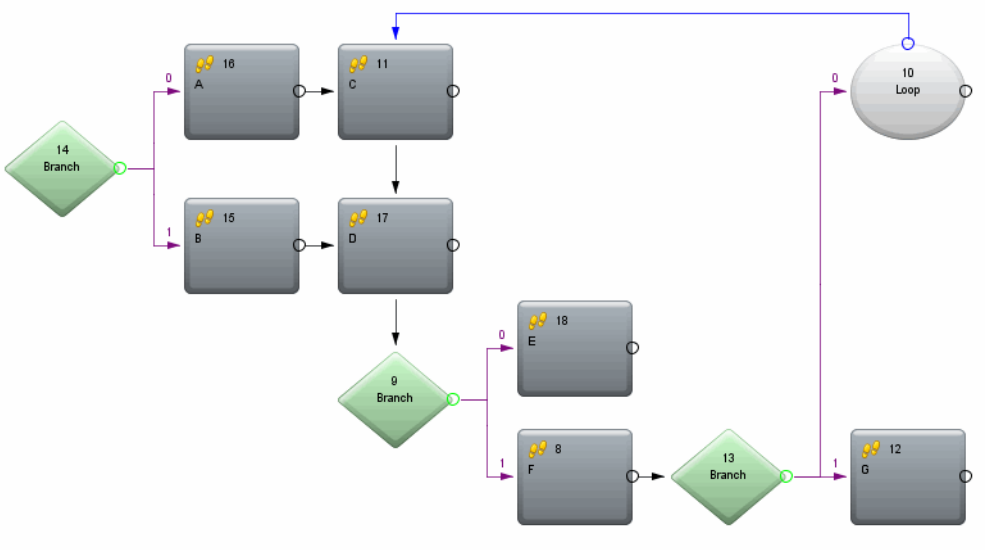
The Discriminator pattern demonstrates how an Or task can be used to wait for tasks to complete before executing the next task in the process. It also shows how a loop can be used to repeat the process.



A discriminator is modeled using an Or task (with the **Cancel unfinished predecessors as soon as any predecessor is done** option cleared), in a Loop. As soon as Task 2, 3, or 4 completes, Task 5 becomes active. The Loop task waits until Tasks 2, 3, and 4 have all completed and then reset the states of all tasks in the Loop body back to ready. Finally, it makes Task 1 active again. The loop does not activate Task 1 until all predecessor tasks (2, 3, 4, and 5) have completed.

Arbitrary Cycles

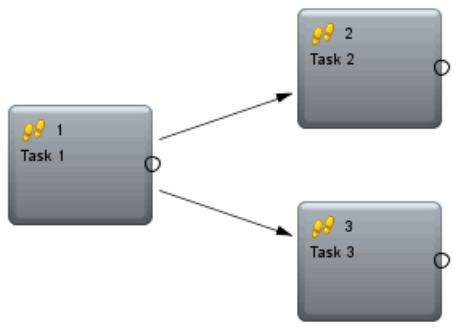
The Arbitrary Cycles pattern demonstrates how a process can be designed for Tasks to be completed repeatedly. The cycle can have multiple entry and exit points.



An arbitrary cycle is modeled using a Loop task. In the preceding example, the Loop ends when conditions set by the branch tasks result in tasks E or G being reached.

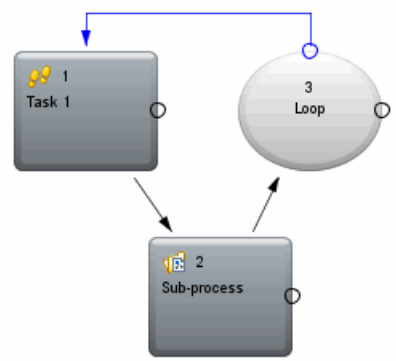
Implicit Termination

Implicit Termination occurs when there are no further tasks to complete. In the following example, the process is complete when tasks 2 and 3 are complete (as they have no successors).



Multiple Instances without Synchronization

This pattern creates multiple instances of a sub process.

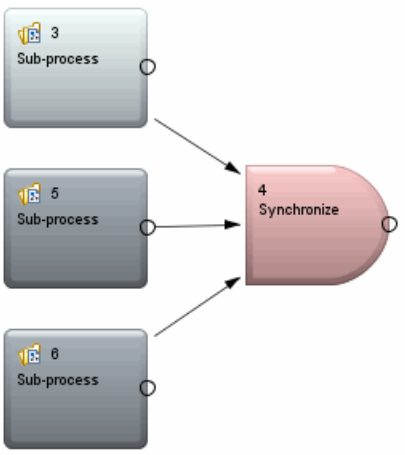


Multiple instances of an activity can be created using a Sub-process task (with **Launch Sub-process Instance as New Top-level Process** selected and **Wait for Sub-process Instance to Complete** selected) in a Loop. The parent process continues without waiting for the sub-process to complete.

Multiple Instances with a Priori Design Knowledge

This pattern creates multiple (3) instances of a sub-process. This pattern is used when the required number of instances is known when the process template is designed.

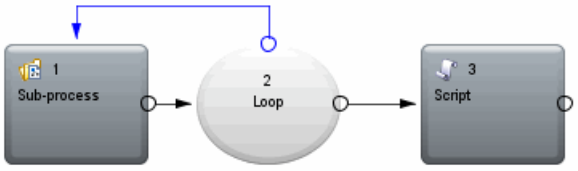
The Sub-processes have **Launch Sub-process Instance as New Top-level Process** and **Wait for Sub-process Instance to Complete** selected. The pattern uses an And task to synchronize the sub-processes (the task following the And task does not execute until all of the sub processes are complete).



Multiple Instances with a Priori Runtime Knowledge

This pattern creates multiple instances of a sub-process. It is used when the required number of instances is not known until the process is executed, but before the instances are created.

A loop is used to create the required number of instances.



The loop `enteringActive` callback is used to increment a counter:

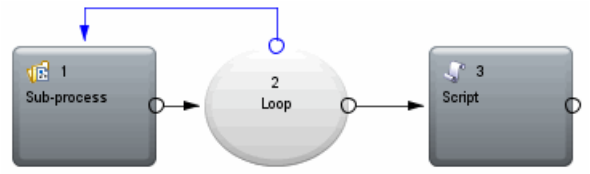
```
long count = ((Number) thisTask.getAttribute("count")).intValue();
thisTask.setAttribute("count", new Long(++count));
thisTask.setDefaultValue(count < 3);
```

The activity to follow the synchronization is modeled as a task with a dependency on the loop task. From its `enteringActive` callback, the synchronization task queries the engine for the child workflow instances created by the Sub-process. If all child instances have not already completed, the synchronization task waits a while and then tries again, until all children have completed. If efficiency is a concern, the synchronization task could register a workflow event listener for each child instance, and be notified when each completes.

```
// get the process manager
ProcessManager pm =
EngineFactory.getDefaultEngine().getProcessManager();

// find all child processes created by the sub-process
// (assumes they have a similar name pattern)
List ids = pm.getProcessesByNamePattern("childOfParent%");

// wait until all children are done
boolean done = false;
while (!done) {
   done = true;
   for (Long id : ids) {
      if
      (!pm.getProcessById(id.longValue()).getState().equals(State.
      DONE)) {
        done = false;
        break;
      }
   }
```

```
   if (!done)
      Thread.sleep(250);
}
```

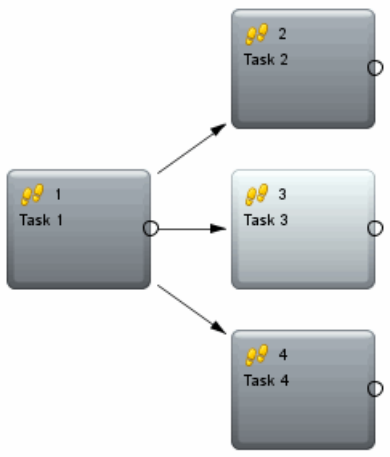
Multiple Instances without a Priori Runtime Knowledge



This pattern is similar to *Multiple Instances with a priori runtime knowledge.* It creates multiple instances of a sub-process.However, it assumes the required number of instances is not known before the first instances are created. In this pattern, the loop condition must be configured so that the loop does not terminate until all of the required instances have been created.

Deferred Choice

The Deferred Choice pattern is similar to the Exclusive Choice pattern. Only one of the successor tasks of Task 1 is executed. However, the choice is not made using a Branch task. Instead, when one of the successor tasks executes, the other tasks are withdrawn.



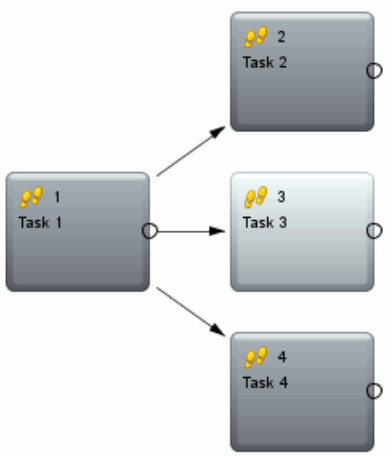
A deferred choice can be implemented by cancelling all other task choices when a user acquires the task. For example, each task choice could have the following code in its `enteringActive` callback:

```
// get predecessor task
```

```
Task predecessor = thisTask.getPredecessors().get(0);

// cancel task choices other than this one
for (Task choice : predecessor.getSuccessors())
   if (choice.getId() != thisTask.getId())
      choice.cancel();
```

Interleaved Parallel Routing

The Interleaved parallel routing pattern executes several tasks. The order in which the tasks are executed is specified at run-time (when the process is executed).



In this example, Task 1 is responsible for determining the order in which its successor tasks are executed. The `enteringActive` callback for Task 1 might use the following code:

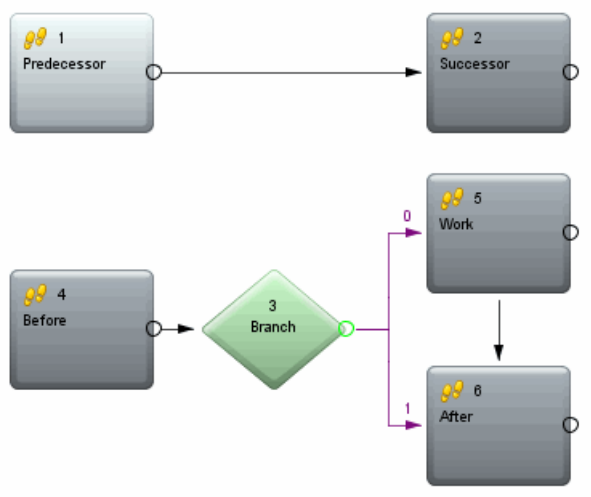
```
// get successor tasks
List successors = thisTask.getSuccessors();
Task successor1 = successors.get(0);
Task successor2 = successors.get(1);
Task successor3 = successors.get(2);

// set dependencies (based on some runtime knowledge)
successor2.addPredecessor(successor1);
successor3.addPredecessor(successor2);
```

If the total order of execution may not be known when the first successor is ready to start, the following method may be used. The `enteringActive` callback for task 1 would determine the first successor, remove the predecessor relationship from it (allowing it to become active), and add a predecessor relationship on it from the other successor tasks. As it determines the order for each other successor, it repeats the process, removing the predecessor relationship and

adding predecessor relationships on all remaining successors. In this way, the successor tasks can be executed as soon as their order is known and all other successors are complete.

Milestone

A milestone is a point in a process where predecessor tasks have completed, but successor tasks have not yet started. If the process is in this state, a Task can be performed.

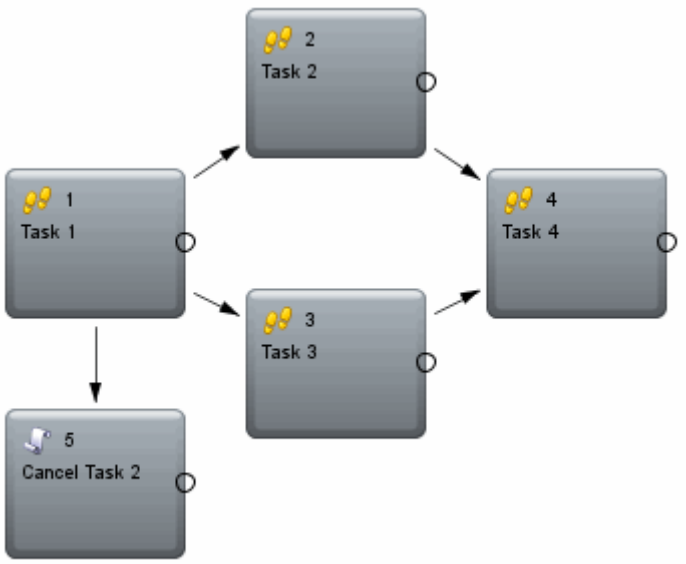


In the pattern, the task labeled `Work` can not execute unless the Predecessor task is complete and the Successor task has not yet started (Successor may have an earliest start date or other dependencies that prevent it from starting as soon as Predecessor finishes). The Milestone is implemented by a branch expression that checks the state of the tasks in question, as follows:

```
not($TaskProp(1,state) = "DONE" and $TaskProp(2,state) = "READY")
```

Cancel Activity

The Cancel Activity pattern demonstrates how a task can be cancelled using script. In the pattern, once task 1 is complete, Tasks 2 and 3 should become active. However, task 2 is cancelled.

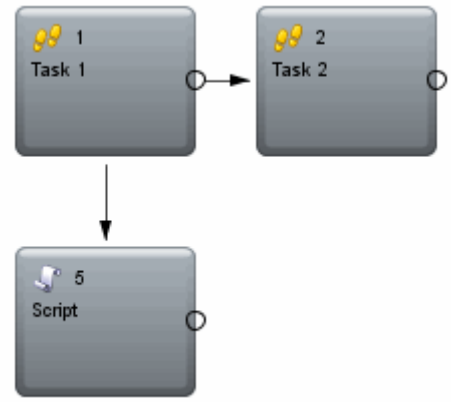


An activity can be canceled using the cancel-task API:

```
// find Task 2
Task task2 = thisProcess.getTaskByName("Task 2");

// cancel Task 2
task2.cancel();
```

Cancel Case

The Cancel Case pattern demonstrates how an entire process instance can be cancelled.



A process instance may be canceled using the `removeProcessInstance` API:

```
Thread work = new Thread() {
   public void run() {
      ProcessManager pm =
      EngineFactory.getDefaultEngine().getProcessManager();
      pm.removeProcessInstance(thisProcess.getId());
   }
};
work.start();
```

CHAPTER 12

APA Expression Language

This section describes the APA Expression Language. For information on the style conventions used in the descriptions of the APA Expression Language, see “Command-line Syntax Conventions” on page 18.

- Supported Functions and Operators
- Expression Language Data Types
- Field References
- Literal Values
- Wildcard References
- Expression Language Operators
- Expression Language Functions

Supported Functions and Operators

Table 1 Supported operators in APA Expression Language

Operator Category	Description
Grouping Operator	“Grouping Operator” on page 177
Logical Operators	“Logical Operators” on page 177
Arithmetic Operators	“Arithmetic Operators” on page 178
Comparison Operators	“Comparison Operators” on page 179
Selection Operator	“Selection Operator” on page 180
String Concatenation Operator	“String Concatenation Operator” on page 180

Table 2 Supported functions in APA Expression Language

Function Category	Description
Boolean	“Boolean Functions” on page 181
Date	“Date Functions” on page 181
Time	“Time Functions” on page 181
String	“String Functions” on page 182
Numeric	“Numeric Functions” on page 183

Expression Language Data Types

The data types supported by the Expression Language are:

- String
- Number
- Date
- Time
- Boolean

The data type constructors described below take the format
<**function name**> (<*argument types*>)

String Data Type

Represents a sequence of Unicode characters.

	Description
`string` `(number)`	Argument is converted as follows: ■ NaN (not a number) is converted to ”NaN” ■ Positive 0 and Negative 0 are converted to ”0” ■ Positive Infinity is converted to ”Infinity”. Negative Infinity is converted to ”-Infinity” ■ Integers are represented in base-10 form with no decimal point and no leading zeros, preceded by a minus sign (-) if the number is negative. ■ Floating point numbers are represented in base-10 form with a decimal point with at least one digit before the decimal and at least one digit after the decimal. Preceded by a minus sign (-) if the number is negative. There must be no leading zeros, unless the leading zero is the only digit before the decimal point. There must be only as many digits after the decimal as are required to uniquely distinguish the number from all other IEEE 754 numeric values.
`string` `(date)`	Returns the argument formatted as `CCYY-MM-DD`.
`string` `(time)`	Returns the argument formatted as `HH:MM:SS` (24-hour)
`string` `(boolean)`	Returns `true` if the argument is true, `false` otherwise.

Number Data Type

Represents an integer or decimal number.

Numeric literals have an optional sign (+ or -) followed by 0 or more digits, followed by an optional decimal point and 1 or more digits. For example: `-1.02`. Incorrectly formatted numeric literals are treated, at evaluation time, as NaN (not a number).

	Description
number (string)	The argument is interpreted according to the rules for numeric literals.
number (date)	Returns the number of days since 1/1/1970
number (time)	Returns the number of seconds since midnight
number (boolean)	Returns 1 if argument is true, 0 otherwise.

Date Data Type

Represents a specific day in history.

	Description
date (string)	Argument must be in the form CCYY-MM-DD. If the argument is not formatted correctly, returns **date**(0).
date (number)	The integral portion of the argument is interpreted as specifying a number of days before or after 1/1/1970.
date (time)	Returns **date**(0)
date (boolean)	Returns **date**(1) if argument is true, otherwise returns **date**(0).

Time Data Type

Represents a point in time during a single day.

	Description
time (string)	Argument must be in the form HH:MM:SS. HH = number of hours (24-hour clock), MM = number of minutes, SS = number of seconds. If the string is not formatted correctly, returns **time**(0).
time (number)	The integral portion of the argument is interpreted as the number of seconds since midnight. If the argument is negative, returns **time**(0).
time (date)	Returns **time**(0).
time (boolean)	Returns **time**(1) if argument is true, otherwise returns **time**(0).

Boolean Data Type

Boolean items have one of two values: true and false.

Boolean literals are not allowed. The functions **`true`**`()` and **`false`**`()` should be used.

	Description
`boolean` `(string)`	Returns true only if the string argument is non-empty. **NOTE** This means that the expression `boolean( string( false() ) )` returns `true`, that is, round-tripping from boolean to string and back is not symmetric.
`boolean` `(number)`	Returns false if the number argument is `+0`, `-0` or not a number, otherwise returns `true`.
`boolean` `(date)`	Returns **`boolean`**`(`**`number`**`(date))`
`boolean` `(time)`	Returns **`boolean`**`(`**`number`**`(time))`.

Field References

Field References return the value of an entry field on a form. For example, `[Entry1]` returns the value of field `Entry1`.

References and Values

Field references are always treated as string values with the following exceptions:

- Numeric format Entry Fields are treated as number values. See “Number Data Type” on page 173.
- Date format Entry Fields are treated as date values. See “Date Data Type” on page 174.
- Time format Entry Fields are treated as time values. See “Time Data Type” on page 174.
- Check Boxes are treated as boolean values. See “Boolean Data Type” on page 175.

If the referenced field is renamed the expression is updated to reflect the name change.

Literal Values

The Expression Language supports two literal values:

- String
- Numeric

There is no direct support for date or time literals. The functions **`date`**`()` and **`time`**`()` should be used to convert string or number literals to the desired type.

Example: `date( "2002-10-23" )` or `time( 60 )`.

There is no direct support for boolean literals. The functions **`true`**`()` and **`false`**`()` should be used.

String literals

String literals are text delimited by double-quotes (for example, "`A String Literal`"). The double-quote character may be included in a string literal by preceding it with a backslash (for example, "`This double-quote character \" will be included in the string literal`".

For more information, see "String Data Type" on page 173. String literals may contain XML-style numbered entities (for example, `䁂`). Numbered entities are converted to Unicode characters when the expression is parsed.

Numeric literals

Numeric literals must conform to the format specified in "Number Data Type" on page 173.

Wildcard References

Certain functions operate on multiple values (for example, `sum`). For example, you might need to sum multiple fields and place the result in another field. Rather than typing each field name, you can use (*) and (?) as wildcard characters in field references.

- * (asterisk). Matches zero or more characters. Example: `[ExtPrice_*]` matches all fields whose names begin with `ExtPrice_`
- ? (question mark). Matches exactly one character. Example: `[Item?]` matches [`Item1`], [`Item2`], but not `[Item]` or `[Item34]`.

Multiple Wildcard Characters

Wildcard characters can be used multiple times in a field reference (for example, `*Price*` is a valid field reference that would select all fields in which `Price` appears in their name). If multiple `*` characters appear next to each other, they are treated as if there were only a single `*` character (for example, `Ext***Price` would select the same fields as `Ext*Price`).

Publish

During publishing, wildcard references are replaced automatically with a list of fields.

Delete Fields

You can rename or delete a field that is referenced in an expression using a wildcard character. When the form is published, the field is ignored.

For Example: A form contains three fields; `Fld1`, `Fld2` & `Fld3`. The expression `sum([Fld*])`expands to `sum([Fld1], [Fld2], [Fld3])`. If `Fld2` is deleted and `Fld3` is renamed to `Total`, the expression expands to `sum([Fld1])`.

Expression Language Operators

The APA Expression language includes a range of operators.

Grouping Operator

You can use parentheses ( ) to group parts of an expression. The grouping operator overrides the built-in precedence rules to affect the order of evaluation in an expression. See the operator descriptions in this section for more information about precedence rules.

Logical Operators

All logical operators evaluate to Boolean values.

Operator	Description
`not`	Evaluates to true only if its operand is false. `not` has the highest precedence of the logical operators. Takes a single Boolean operand. The operand is implicitly converted to a boolean as if by a call to **`boolean()`**. NOT associates with the operand to its right.
`and`	Evaluates to `true` only if both of its operands are true. Takes two Boolean operands. The operands are implicitly converted to `Boolean` values as if by a call to **`boolean`**`()`.
`or`	Evaluates to `true` if either of its operands are true. `or` has the lowest precedence of the logical operators. Takes two Boolean operands. The operands are implicitly converted to `Boolean` values as if by a call to **`boolean`**`()`.

Arithmetic Operators

The Expression Language uses the following arithmetic operators.

Multiplicative Operators

The multiplicative operators are:

Operator	Description
*	multiply
/	divide

- The multiplicative operators have higher precedence than the additive operators.

Additive Operators

The additive operators are:

Operator	Description
+	add
-	subtract

- The additive operators have lower precedence than the multiplicative operators.
- Arithmetic operators with same precedence are evaluated left-to-right.
- Arithmetic operators always convert their operands to number as if by a call to **`number`**`()`.
- Arithmetic operators always evaluate to a number.

Comparison Operators

The Expression Language uses the following equality and relational operators:

Equality Operators

The equality operators are:

Operator	Description
=	equal
!=	not-equal

Relational Operators

Operator	Description
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to

Precedence

- Equality operators have higher precedence than the relational operators.

- All equality and relational operators with the same precedence are evaluated left-to-right.

Evaluation

All equality and relational operators evaluate to Boolean values.

Conversion

If the left operand is a `string` value, the right operand is implicitly converted to a `string` value and the strings are compared lexicographically.

Otherwise, the operands are implicitly converted to `number` and then compared.

Selection Operator

The selection operator `if` takes three operands. The first operand is implicitly converted to *Boolean* as if by a call to **`Boolean()`**. If this operand evaluates to `true`, the operator returns the value of its second operand, otherwise it returns the value of its third operand.

The selection operator evaluates to the type of the second operand (the third operand is implicitly converted to the type of the second operand).

Example

```
if([AGI] < 64000, [AGI] * 0.076, 4864)
```

This means if the expression `[AGI] < 64000` is true, return the result of `[AGI] * 0.076`, otherwise return the value `4864`.

String Concatenation Operator

The string concatenation operator (&) evaluates to a string.

The string concatenation operator implicitly converts each of its arguments to string as if by a call to **`string()`**.

The expression `string & string` is equivalent to calling the function **`concat()`**, that is, `concat(string,string)`.

Expression Language Functions

Every function has a well-defined prototype that defines its return type and the number and types of arguments that it takes. An asterisk (*) following an argument indicates that there may be zero or more instances of the argument.

All functions implicitly convert their arguments to the required type as if by a call to the appropriate type conversion function.

Boolean Functions

The Expression Language uses these Boolean functions:

Function	Description
`boolean` **`true`**`()`	Returns the Boolean value `true`.
`boolean` **`false`**`()`	Returns the Boolean value `false`.
`boolean` **`not`**`(boolean` *`b`*`)`	Returns the logical negation of *b*. `not` evaluates to `true` if and only if *b* is `false`. Takes a single Boolean operand. The operand is implicitly converted to a Boolean as if by a call to **`Boolean`**().

Date Functions

Function	Description
`date` **`today`**`()`	Returns the current date
`number` **`year`**`(date` *`d`*`)`	Returns a number representing the year component of *d*.
`number` **`month`**`(date` *`d`*`)`	Returns a number representing the month component of *d*.
`number` **`day`**`(date` *`d`*`)`	Returns a number representing the day component of *d*.

Time Functions

Function	Description
`time` **`now`**`()`	Returns the current time.
`number` **`hour`**`(time` *`t`*`)`	Returns a number representing the hour component of *t* (0–23).
`number` **`minute`**`(time` *`t`*`)`	Returns a number representing the minutes component of *t* (0–59).
`number` **`second`**`(time` *`t`*`)`	Returns a number representing the seconds component of *t* (0–59)

String Functions

Function	Description
`number` **`length`**`(string` *`S`*`)`	Returns the length (number of characters) of *S*.
`string` **`trim`**`(string` *`S`*`)`	Removes leading and trailing white-space from *S*. *S* itself is not modified.
`string` **`normalize-space`**`(string` *`S`*`)`	Removes leading and trailing white-space from *S* and replaces sequences of space characters with single space characters. *S* itself is not modified.
`string` **`left`**`(string` *`S`*`, number` *`N`*`)`	Returns the first *N* characters from *S*. *S* itself is not modified.
`string` **`right`**`(string` *`S`*`,number` *`N`*`)`	Returns the last *N* characters from *S*. *S* itself is not modified.
`string` **`substring`**`(string` *`S`*`,number` *`start`*`,number` *`count`*`)`	Returns *`count`* characters, starting at position *`start`* in *S*. *S* itself is not modified.
`string` **`upper`**`(string` *`S`*`)`	Converts *S* to upper case. *S* itself is not modified.
`string` **`lower`**`(string` *`S`*`)`	Converts *S* to lower case. *S* itself is not modified.
`string` **`concat`**`(string, string*)`	Returns a string produced by concatenating the arguments. May take wildcard field references.
`boolean` **`contains`**`(string` *`str`*`, string` *`sub`*`)`	Returns `true` if *`str`* contains *`sub`*, and otherwise returns `false`.
`boolean` **`starts-with`**`(string` *`str`*`, string` *`prefix`*`)`	Returns `true` if *`str`* starts with `prefix`, and otherwise returns `false`.
`string` **`substring-before`**`(string` *`str`*`, string` *`sub`*`)`	Returns the substring of *`str`* that precedes the first occurrence of *`sub`* in *`str`*, or an empty string if *`str`* does not contain *`sub`*. For example, `substring-before("1999/04/01","/")` returns `1999`.
`string` **`substring-after`**`(string` *`str`*`, string` *`sub`*`)`	Returns the substring of *`str`* that follows the first occurrence of *`sub`* in *`str`*, or an empty string if *`str`* does not contain *`sub`*. For example, `substring-after("1999/04/01","/")` returns `04/01`, and `substring-after("1999/04/01","19")` returns `99/04/01`.

Numeric Functions

Function	Description
number **sum**(*number, number**)	Returns the sum of the arguments. Can take wildcard field references.
number **avg**(*number, number**)	Returns the average of the arguments. Can take wildcard field references.
number **min**(*number, number**)	Returns the minimum of the arguments. Can take wildcard field references.
number **max**(*number, number**)	Returns the maximum of the arguments. Can take wildcard field references.
number **count-non-empty**(*string, string**)	Returns the number of non-empty fields. Can take wildcard field references.
number **floor**(*number N*)	Returns the largest (closest to positive infinity) integer that is not greater than *N*.
boolean **is-number**(*string*)	Returns `true` if *data* contains a valid number. If `string` is empty, the function returns `false`.
number **ceiling**(*number N*)	Returns the smallest (closest to negative infinity) integer that is not less than *N*.
number **round**(*number N*)	Returns the integer that is closest to *N*. If there are two such numbers, then the one that is closest to positive infinity is returned. If *N* is NaN, then NaN is returned. If *N* is positive infinity, then positive infinity is returned. If *N* is negative infinity, then negative infinity is returned. If *N* is positive zero, then positive zero is returned. If *N* is negative zero, then negative zero is returned. If *N* is less than zero, but greater than or equal to -0.5, then negative zero is returned. **NOTE** The result of calling the round function is not the same as the result of adding 0.5 and then calling the `floor` function.

Task Functions

Function	Description
`string` **`selected-submit-action`**`(taskID)`	Used in Process Studio. Retrieves the value component (as a string) of the selected submit action for the given task. The `taskId` parameter must refer to a scoped task identifier on the current process template. Returns an empty string if the task identifier is invalid or if the selected submit action has not been set. For example: selected-submit-action(12) – returns the submit action value of the task with scoped ID 12.

PART 4 Process Studio Tutorial

This section contains a tutorial that demonstrates how to create a process using Process Studio.

- Getting Started
- Process Studio Tutorial

CHAPTER 13

Getting Started

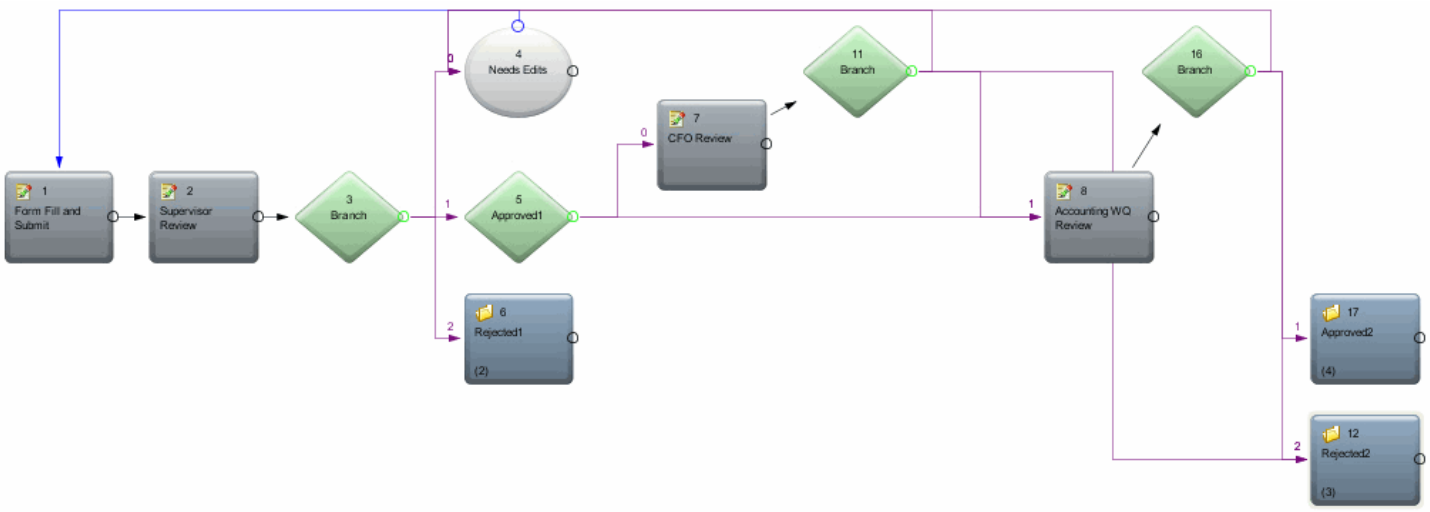
This section provides an overview the Process Studio Tutorial and describes prerequisites.

- Tutorial Overview
- Tutorial Prerequisites

Tutorial Overview

In this tutorial, you will create a simple `Purchase Order` process template in Process Studio. The completed process is shown in Figure 3:

Figure 3 Completed Tutorial Process



The `Purchase Order` process starts with a `Purchase Order` form that is submitted by the process originator. The form is then reviewed by the originator's supervisor (and the Chief Financial Officer, if the purchase order is $10,000 or greater). The form is then sent to the Accounting Work Queue for final approval.

Tutorial Prerequisites

Before starting the tutorial, ensure the following are complete:

- The user relationships and work queues that are used in this tutorial are configured on the APA Server by an Administrator. See "Create Relationships and Work Queues" on page 189.
- The Purchase Order form is available in the `TutorialForms` folder on the APA server. If the Purchase Order form is not available, it must be published. See "Publish the Purchase Order Form" on page 191.

Create Relationships and Work Queues

Some tasks in the `Purchase Order` process are assigned to Relationships and Work Queues. To complete the tutorial, these must be available on the APA Server.

NOTE If it is not possible to have the Relationships and Work Queues set up on your APA server, you can assign all tasks to **Process Originator** and skip the applicable steps.

Add Relationships

For the purpose of this tutorial, an APA Administrator should add a `Tutorial_Supervisor` Relationship and `Tutorial_CFO` (Chief Financial Officer) Relationship to the APA Server.

NOTE Only Administrators or Relationship Administrators can configure Relationships.

To add a relationship

1. Start the Management Console.
2. In the navigation pane, click **Relationships**.
3. On the **File** menu, click **Add**.

   The **User Relationship** dialog box opens.
4. In the **Relationship ID** box, type **`Tutorial_Supervisor`**.
5. Select the **Allow users to modify** check box.
6. Click **Display Settings**.

   The **Display Settings** dialog box opens.
7. Double-click the **Name** cell in the `English` language row (or another language row that you want to use), and type **`Tutorial_Supervisor`**, and then press ENTER.
8. Double-click the **Description** cell and type **`Tutorial_Supervisor`**, and then press ENTER.
9. Click **OK**. Click **OK** again.

   A message is displayed asking whether you want to Customize the relationship. This allows you to assign users to the relationship.
10. Click **Yes**.

The **User Relationship** dialog box opens.

11. Click the **By Target Relationship** tab.
12. In the **Tutorial_Supervisor** area, click **Search**.

    The **Browse for Relationship target** dialog box opens.
13. Search for and select the user you want to act as the Supervisor. During the tutorial, this user receives forms for approval. Click **OK**.
14. In the **Users** area, click **Search**.
15. Search for and select all the users who might participate in this process as the process originator.
16. Click **OK**.

Repeat these steps to create a `Tutorial_CFO` Relationship.

Refer to the *Management Console User Guide* for more information about configuring and using Relationships.

Add a Work Queue

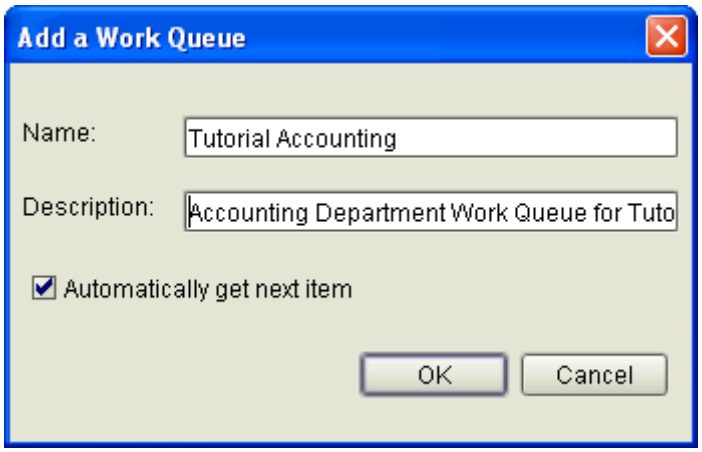
NOTE Only Administrators can configure Work Queues.

For the purpose of this tutorial, an APA Administrator should add a `Tutorial_Accounting` Work Queue to the APA Server.

To add a work queue

1. Start the Management Console.
2. In the navigation pane, click **Work Queues**.
3. On the **File** menu, click **Add**.

   The **Add a Work Queue** dialog box opens.
4. In the **Name** box, type **`Tutorial_Accounting`**.
5. In the **Description** box, type **`Accounting Department Work Queue for Tutorial`**.



6. Verify that the **Automatically get next item** check box is selected.
7. Click **OK**. The `Tutorial_Accounting` Work Queue is created.
8. Click the Tutorial_Accounting Work Queue and click **View** > **Permissions**.

   The **Work Queue Permissions** dialog box opens.
9. Click **Search Users** to search for and assign users to the Work Queue. During the tutorial process, these users are sent forms for approval.

NOTE Only individuals assigned the Process Approver role can be assigned to a Work Queue.

NOTE Refer to the *Management Console User Guide* for more information about Work Queues.

10. Click **OK**.

Publish the Purchase Order Form

For the purposes of this tutorial, a `Purchase_Order` form has already been published to the APA server and is available in the `TutorialForms` folder.

If the `Purchase_Order` form is not available, a user assigned the Form Publisher role can publish it to the APA server.

To publish the Purchase_Order form

1. Open the `Purchase_Order` form in Form Designer. This form is available in the `Tutorial` folder in the Form Designer directory (default location: `C:\ProgramFiles\Cardiff\LiquidOffice\FormDesigner\Tutorial`).
2. On the **File** menu, click **Publish**.

3. Accept the default settings on the **General Properties**, **Permissions**, and **Routing Settings** pages of the **Publishing Wizard** dialog box
4. On **Data Exports** page, select the **Export data using the My Data Client** check box and accept the remaining default settings.
5. Click **Finish** and **Done**.

   The form is published to the APA server. Refer to the *Form Designer User Guide* or *Management Console User Guide* for more information about publishing forms to the APA server.

CHAPTER 14

Process Studio Tutorial

This section describes the steps to create a simple `Purchase Order` process template in Process Studio.

Before starting the tutorial, ensure you set up the relationships and work queues and Purchase Order form on the APA server (see “Tutorial Prerequisites” on page 188).

In this tutorial, you will work through the following tasks:

- Create a Process Template and Add Tasks
- Create the Supervisor Review
- Add a Branch Task and Successor Tasks
- Configure the Loop Task
- Define the Approved Path Using an Expression
- Configure the Rejected Path
- Configure the CFO Review Task
- Configure the Accounting Work Queue Review Task
- Configure the CFO Review Branch Task
- Configure the Accounting Review Branch Task
- Configure the Rejected2 Container
- Configure the Approved Container

Create a Process Template and Add Tasks

This section of the tutorial describes how to:

- Create a New Process Template
- Add a Form Task to the Template
- Configure the Form Task
- Save the Process Template

Create a New Process Template

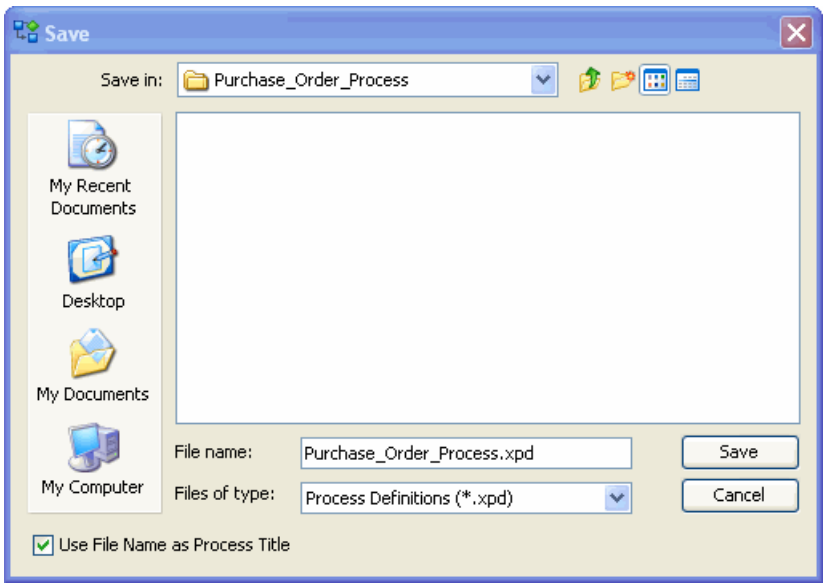
Use the following procedure to create a new process template.

To create and save a process template

1. Start Process Studio.
2. On the **File** menu, click **New**.

   A new process template is created.
3. On the **File** menu, click **Save As**.

   The **Save** dialog box opens.



4. Navigate to the location where you want to save the process template.
5. In the **File Name** box, type **`Purchase_Order_Process`**.

6. Click **Save**.

Related Topics

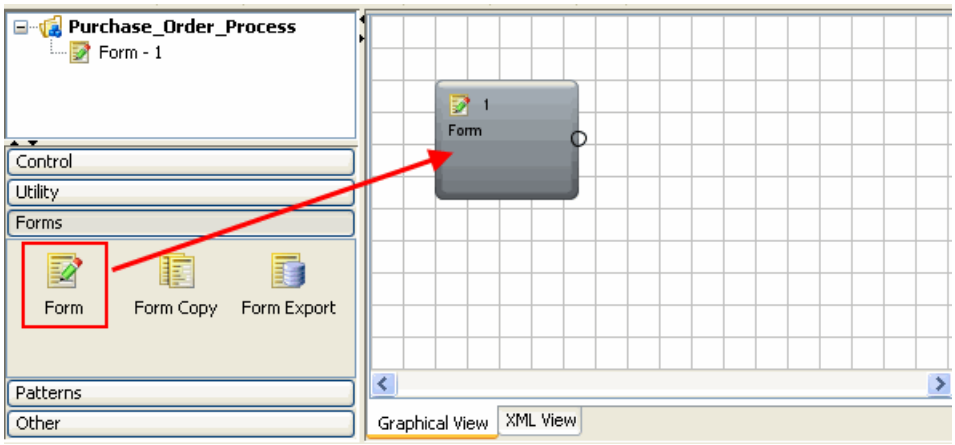
- “Create a New Process” on page 46
- “Save As” on page 48

Add a Form Task to the Template

The first task (the *start* task) in the process is a *Form task*. Use the following procedure to add a Form task to the process template.

To add a Form task to the process template

- Drag a Form task from the **Task Library** (**Forms** drawer) to the process template **Graphical View**).



Related Topics

- “Add a Task” on page 81
- “Form Task” on page 77

Configure the Form Task

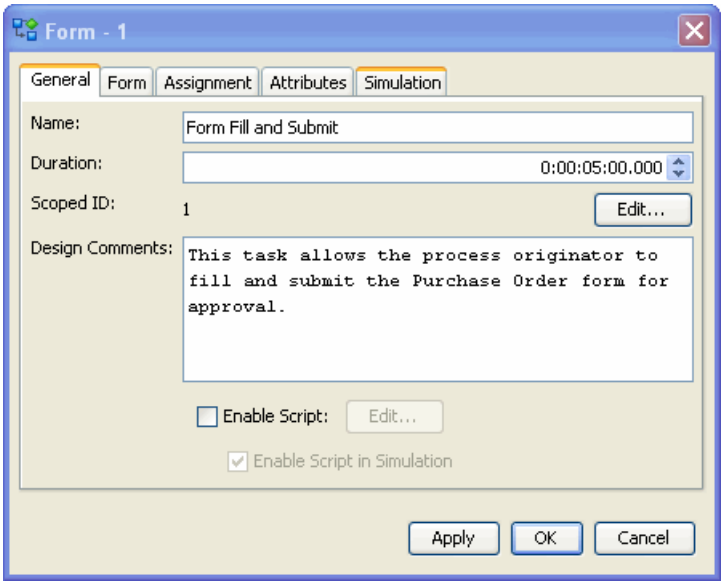
Use the following procedure to:

- apply a name, duration, and design comments to the task.
- assign a published form to the task.
- assign the task to a user. For this task the assigned user is the individual who initiates the `Purchase Order` process on the APA server (the *process originator*).

To configure the Form task

1. Double-click the Form task to edit its properties.

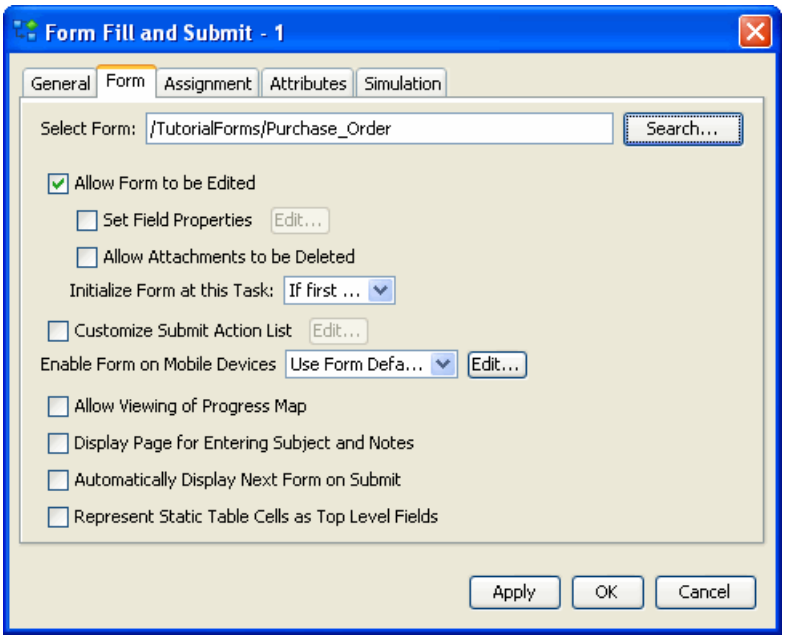
   The task's properties dialog box opens.



2. In the **Name** box, type **`Form Fill and Submit`**.
3. In the **Duration** box, type or select **`0:00:05:00.000`** (5 minutes).
4. In the **Design Comments** box, type the following text:

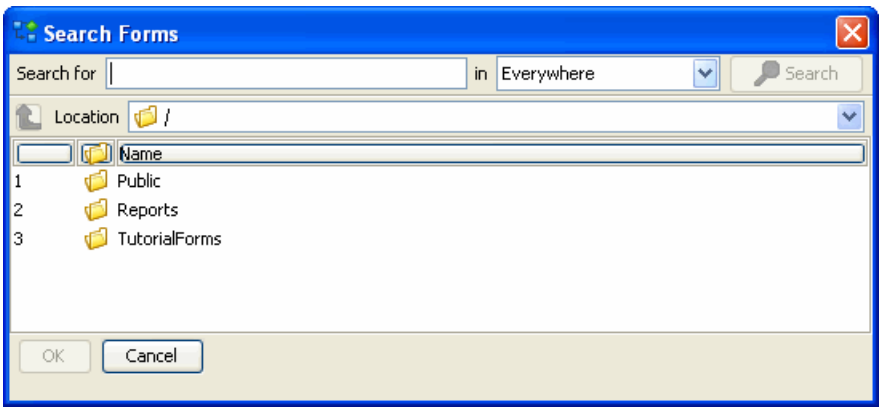
   `This task allows the process originator to fill and submit the Purchase Order form for approval.`

5. Click the **Form** tab.

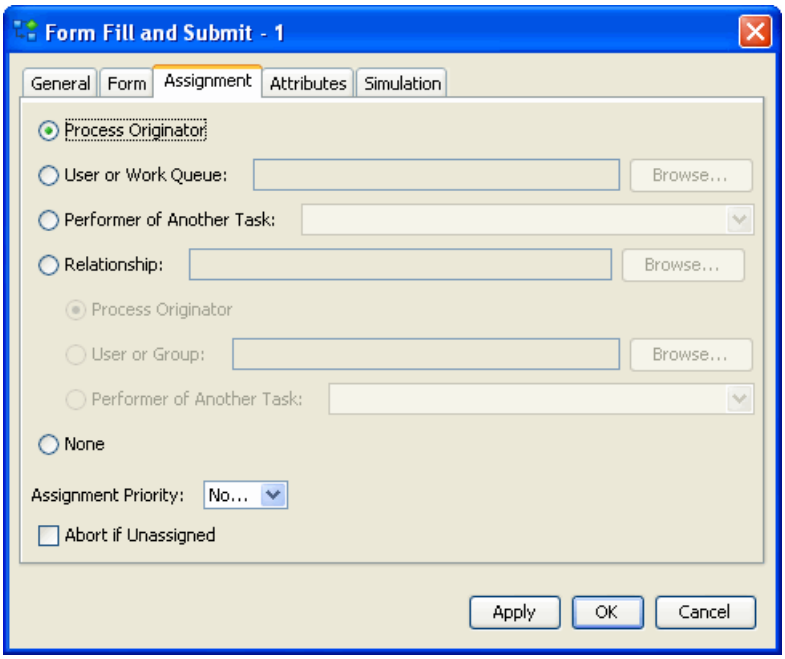


6. Click **Search**.

   The **Search Forms** dialog box opens.



7. Select the `Purchase_Order` form in the `TutorialForms` folder, and click **OK**.
8. Select the **Allow Form to be Edited** check box.
9. Click the **Assignment** tab.



10. Click **Process Originator**.
11. Click **OK**.

Related Topics

- “Configure Tasks” on page 91
- “Configure General Properties” on page 92
- “Configure a Form Task” on page 113
- “Task Assignment” on page 95

Save the Process Template

Use the following procedure to save the process template.

To save the process template

- On the **File** menu, click **Save**.

Related Topics

- “Save a Process” on page 48

Create the Supervisor Review

The second task in the `Purchase Order` process is a *Form* task. This task is assigned to the process originator's supervisor. The supervisor receives the completed `Purchase Order` form in their Inbox. The supervisor reviews the form and signs it. They can then choose to approve the form, reject it, or send it back to the originator to be edited.

This section of the tutorial describes how to:

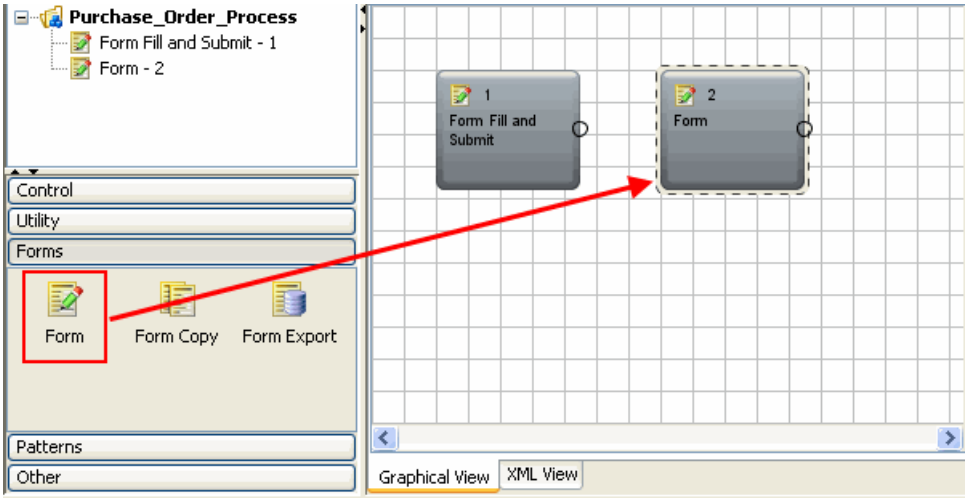
- Add a Second Form Task to the Process Template
- Configure the Second Form Task (set form field properties, and add custom submit actions)
- Assign the Form Task Using a Relationship

Add a Second Form Task to the Process Template

Use the following procedure to add a second form task to the process template.

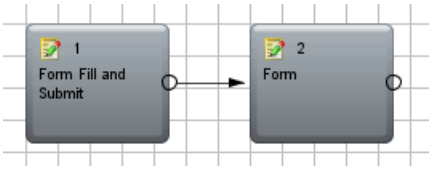
To add a second form task

1. Drag a Form task from the **Task Library** onto the process template.



2. Connect the `Form Fill and Submit` task to the second Form task.

   To connect the tasks, drag the black connector from the right side of the first form task onto the second form task.



Related Topics

- “Add a Task” on page 81
- “Form Task” on page 77
- “Connect Tasks” on page 83

Configure the Second Form Task

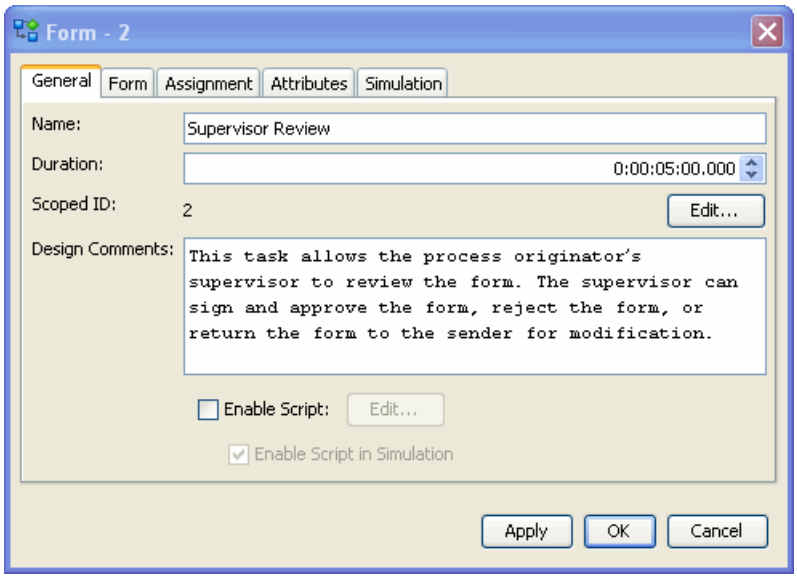
Use the following procedure to:

- configure the general properties (name, duration, and design comments) of the form task.
- assign a published form to the task.
- modify the properties of the form fields so that the supervisor cannot edit the form (only sign it).
- customize the submit actions available on the form. Custom submit actions allow the supervisor to choose what to do with the form (approve it, reject it, or return it to the originator for editing).
- configure the form to display a notes page on submission. The supervisor can then type notes to explain why the form has been rejected or returned for editing.

To configure the general properties of the Form task

1. Double-click the new Form task to edit its properties.

   The task’s properties dialog box opens.

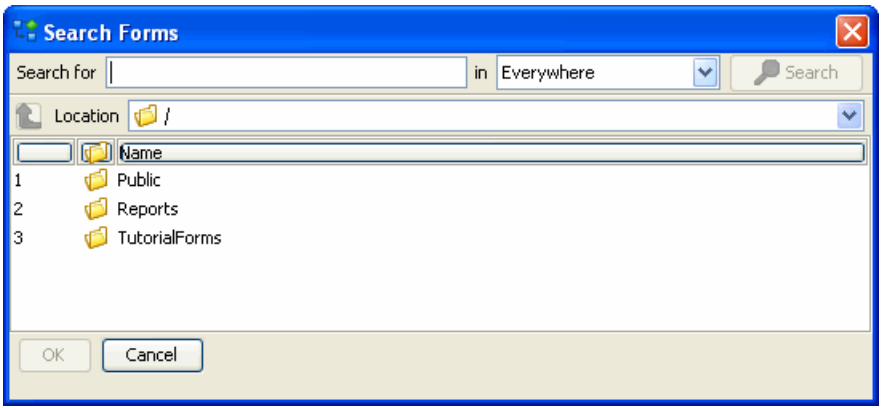


2. In the **Name** box, type **`Supervisor Review`**.
3. In the **Duration** box, type or select **`0:00:05:00.000`** (5 minutes).
4. In the **Design Comments** box, type the following text:

   `This task allows the process originator's supervisor to review the form. The supervisor can sign and approve the form, reject the form, or return the form to the sender for modification.`

5. Click the **Form** tab.
6. Click **Search**.

   The **Search Forms** dialog box opens.

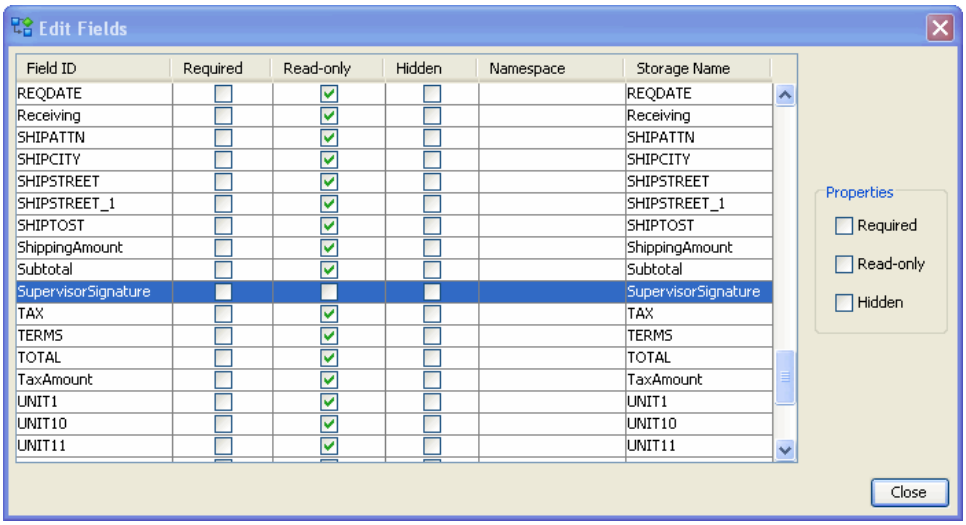


7. Click the `Purchase_Order` form in the `TutorialForms` folder, and click **OK**.

NOTE The `Purchase Order` form must be available on the APA server. If the form is not published to the APA server, a Form Publisher can publish it. See "Publish the Purchase Order Form" on page 191.

8. Select the following check boxes:
   - **Allow Form to be Edited**
   - **Allow Viewing of Progress Map**
   - **Display Page for Entering Subject and Notes**
   - **Automatically Display Next Form on Submit**
9. Select the **Set Field Properties** check box and click **Edit**.

   The **Edit Fields** dialog box opens. The list of fields is built dynamically by the APA server.



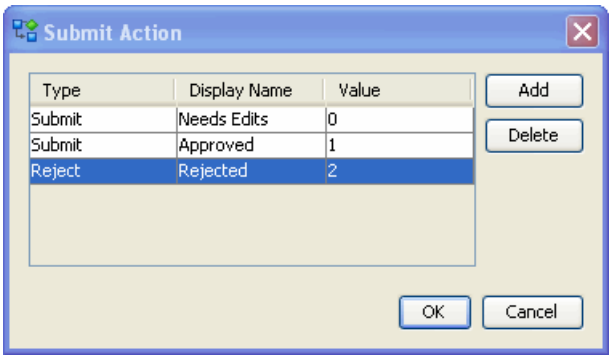
10. Set all of the fields (except `SupervisorSignature`) to Read Only. The easiest way to do this is to select all of the fields, and then select the **Read Only** check box in the **Properties** area. Then select the `SupervisorSignature` row and clear the **Read Only** check box.

TIP To select multiple fields in a range, click the first field and then hold down SHIFT while you click the last field in the range. To select multiple fields that are not contiguous, hold down CTRL while you click each field.

11. Click **Close**.

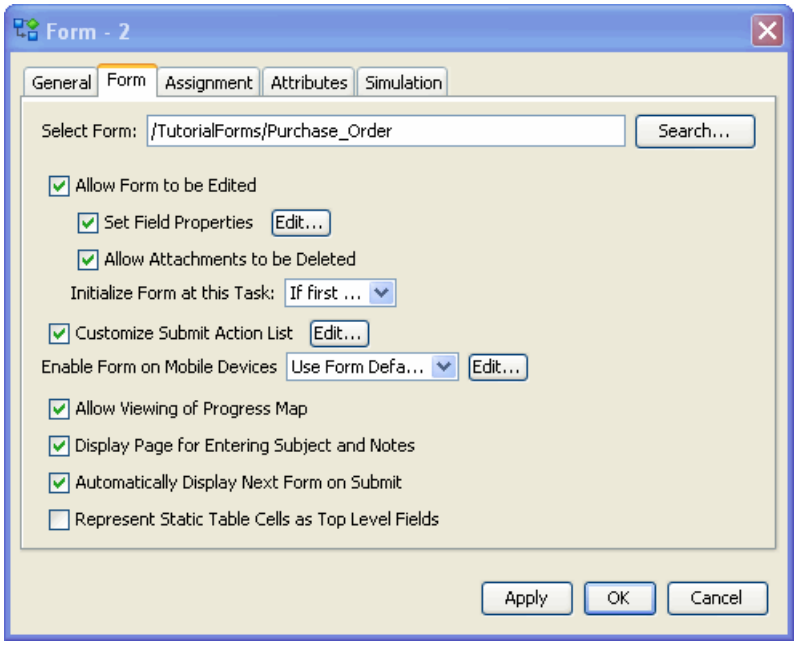
12. Select the **Customize Submit Action List** check box and click **Edit**.

    The **Submit Action** dialog box opens.



13. Configure the custom submit actions for the form as shown in the previous image. Click **Add** to add custom submit actions. Double-click a cell to type or select a value.
14. Click **OK**.

    The **Form** tab should now be configured as shown in the following image:



15. Click **OK**.

Related Topics

- "Configure Tasks" on page 91

- “Configure General Properties” on page 92
- “Configure a Form Task” on page 113
- “Search the APA Server” on page 132

Assign the Form Task Using a Relationship

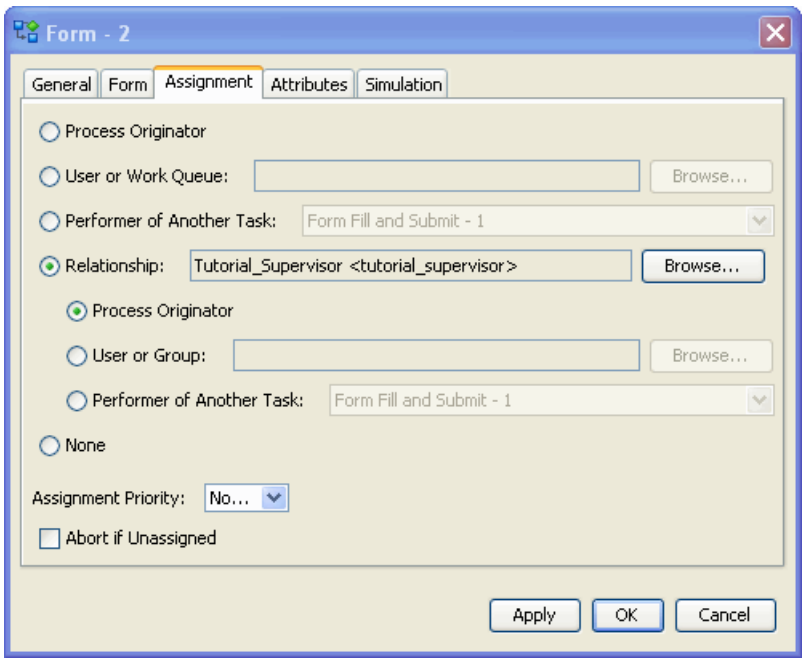
Use the following procedure to assign the form to a supervisor using the `Tutorial_Supervisor` relationship. Using a relationship ensures that the form is sent to the correct supervisor for approval, regardless of the identity of the process originator.

To assign the task to a Supervisor

1. Double-click the `Supervisor Review` form task to edit its properties.

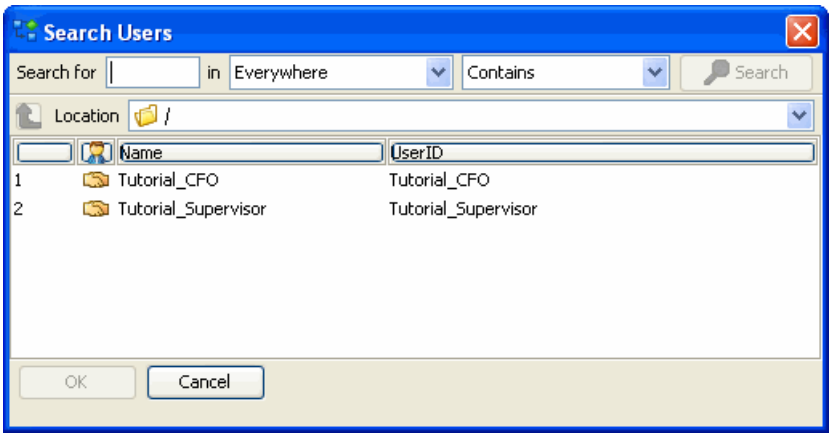
   The task’s properties dialog box opens.

2. Click the **Assignment** tab.



3. Click **Relationship** and click **Browse**.

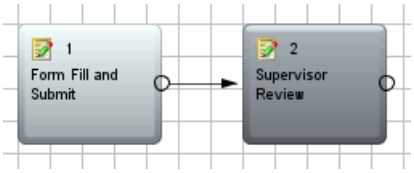
   The **Search Users** dialog box opens.



4. Click the **Tutorial_Supervisor** Relationship and click **OK**.
5. Click the **Process Originator** relationship type.

   The task is assigned to the supervisor of the process originator.
6. Click **OK**.

   The process should look similar to the following diagram:



Related Topics

- “Task Assignment” on page 95
- “Search the APA Server” on page 132

Add a Branch Task and Successor Tasks

After the `Supervisor Review` form task, a branch task is used to select a process path based on the action selected by the supervisor.

This section of the tutorial describes how to:

- Add a Branch Task to the Process.
- Add Successor Tasks to the Branch Task.
- Configure the Branch Task.

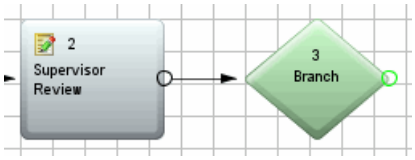
Add a Branch Task to the Process

Use the following procedure to add a branch task to the process.

To add the Branch task

1. Add a Branch task to the process (drag a branch task from the **Task Library** onto the **Graphical View**).
2. Connect the `Supervisor Review` task to the Branch task.

   Drag the connector on the right side of the `Supervisor Review` task onto the Branch task.



Related Topics

- “Add a Task” on page 81
- “Branch Task” on page 74
- “Connect Tasks” on page 83

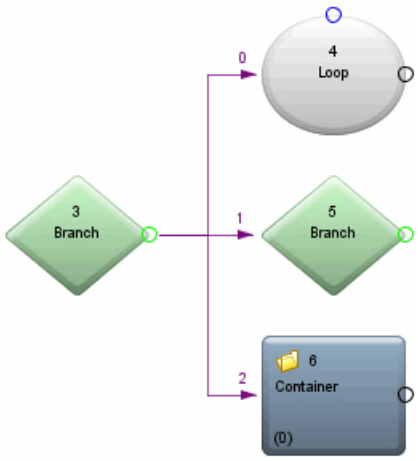
Add Successor Tasks to the Branch Task

Before the Branch Task can be configured, you must add its successor tasks to the process.

To add successor tasks to the process

1. Add a *Loop task*, another *Branch* task, and a *Container* task to the process.
2. Connect the Branch task to its successor tasks as shown in the following diagram.

   To connect the Branch task, drag the green connector from the right side of the branch task onto the loop task. When you release the mouse button, the path is drawn. Repeat this process, dragging the connector onto the second branch task, and then the container task.



Related Topics

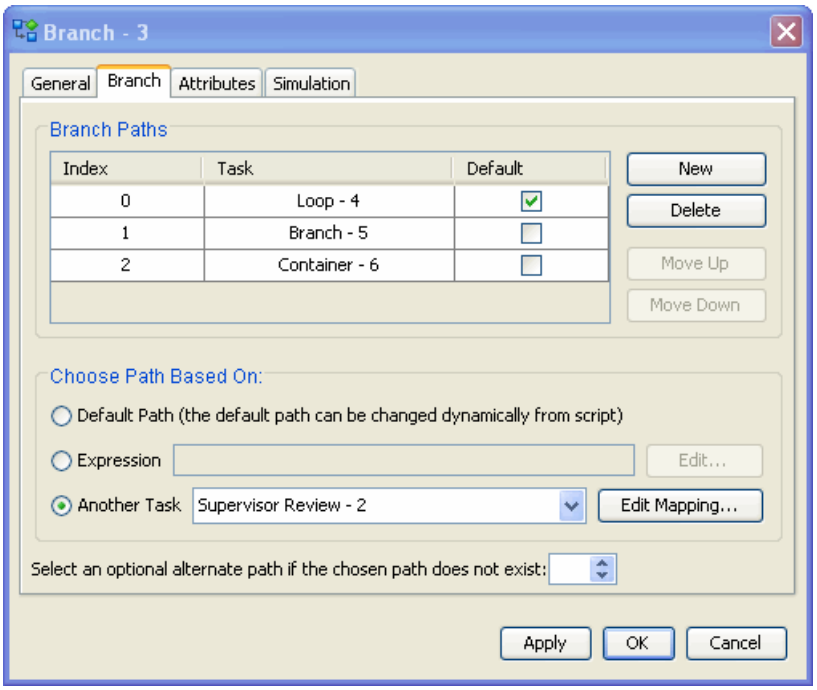
- “Add a Task” on page 81
- “Connect a Branch Task” on page 84

Configure the Branch Task

Use the following procedure to configure the branch Task.

To configure the Branch Task

1. Double-click the first Branch task.
2. Click the **Branch** tab.



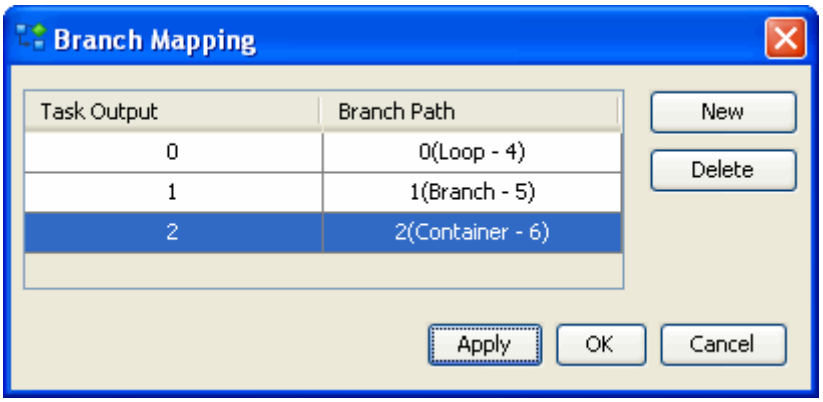
3. In the **Branch Paths** area, the successor tasks are assigned Index values. If the path Index values for your branch task do not match those in the preceding image, select a row in the **Branch Paths** table and click **Move Up** or **Move Down**.

   Ensure the "0" path index points to the Loop task, the "1" path index points to the second Branch task, and the "2" path index points to the Container task.

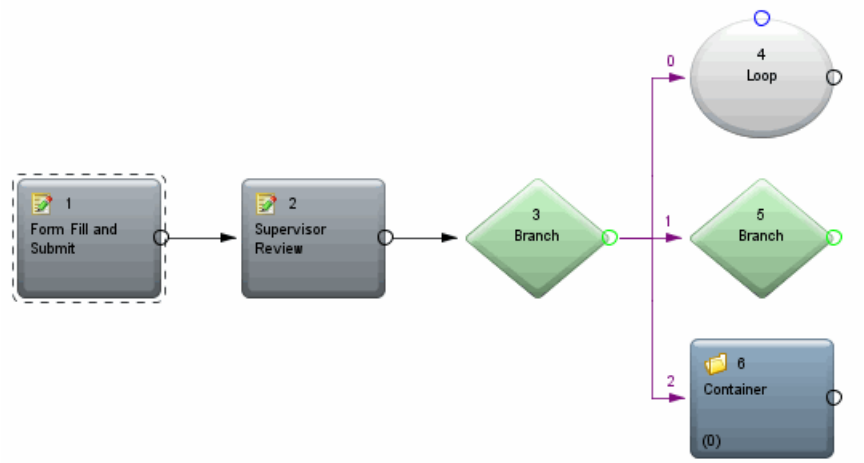
   The path index values are important because the branch paths are configured to execute based on the submit action selected in the `Supervisor Review` task.

4. In the **Choose Path Based On** area, click **Another Task**.
5. In the **Another Task** list, click **Supervisor Review**.
6. Click **Edit Mapping**.

   The **Branch Mapping** dialog box opens. This dialog box is used to define the relationship between the output of the `Supervisor Review` task and the branch paths.



7. Click **New** three times to add new branch mapping entries to the list.
8. Configure the **Task Output** and **Branch Path** settings as shown in the preceding diagram. Double-click a table cell to edit a value.
9. Click **OK**.
10. Click **OK** again.
11. Save the process. The process should look similar to the following diagram:



Related Topics

- “Configure a Branch Task” on page 100

Configure the Loop Task

This section of the tutorial describes how to:

- Configure the Loop task

The purpose of the Loop task is to return the form to the Process Originator. The Loop task is required if the Supervisor, Chief Financial Officer, or Accounting Work Queue submits the form using the `Needs edits` custom submit action.

The `Purchase_Order` process that you are creating uses Branch Tasks to control the execution of the process. Therefore the loop can be configured to execute based on a default value of True (the loop will execute every time its process path is selected).

Configure the Loop task

Use the following procedure to configure the Loop task.

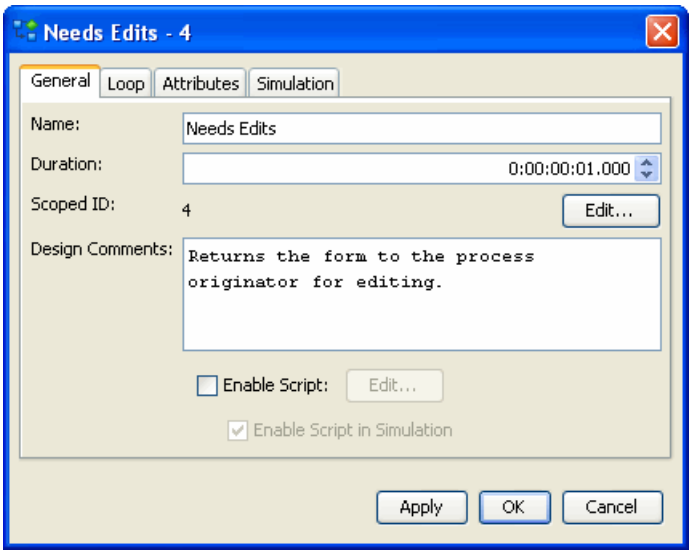
To configure the Loop task

1. Connect the Loop task to the `Form Fill and Submit` task. The `Form Fill and Submit` task is the *target* task (the first task inside the loop).

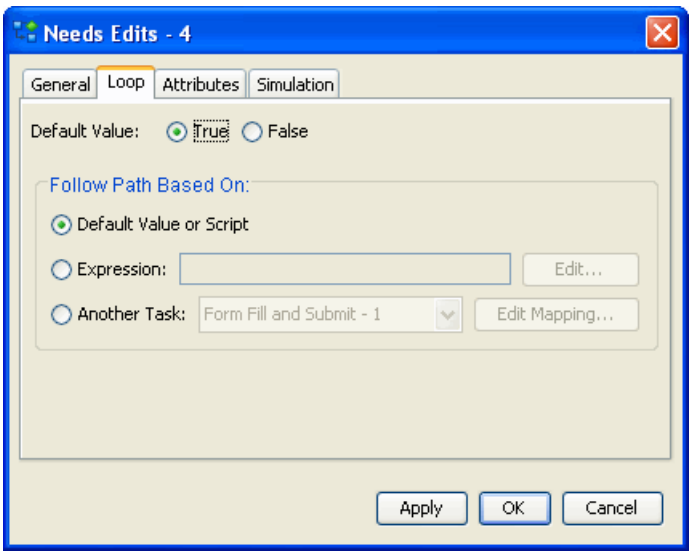
   To connect a loop task to the target task, drag the blue connector from the top of the loop task onto the target task. Do not use the black connector.

2. Double-click the Loop task to edit its properties.

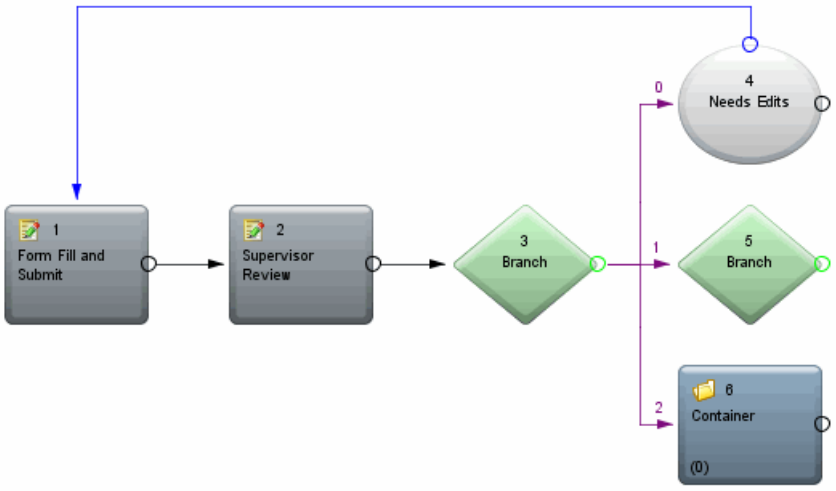
   The properties dialog box opens.



3. In the **Name** box, type **`Needs Edits`**.
4. In the **Design Comments** box, type **`Returns the form to the process originator for editing`**.
5. Click the **Loop** tab.



6. Click **True** to set the **Default Value** to **True**.
7. Click **OK**.
8. Save the process.



Related Topics

- “Loop Task” on page 74
- “Connect a Loop Task” on page 85
- “Configure Tasks” on page 91
- “Configure General Properties” on page 92
- “Configure a Loop Task” on page 102

Define the Approved Path Using an Expression

If the supervisor approves the form submitted by the process originator, the first branch task selects path “1” and the second branch task is executed.

The second Branch task is used to route the form. If the total value of the purchase order (specified by the `Total` field on the `Purchase Order` form) is $10,000 or greater, the form is sent to the Chief Financial Officer (CFO) for approval.

If the total value is less than $10,000, the approval of the CFO is not required. The form is routed directly to the `Tutorial_Accounting` Work Queue for approval.

The Branch task needs to be configured to select a path using an expression. Expressions return a value of 0 (zero) if they evaluate to False, or 1 (one) if they evaluate to true.

The branch path pointing to `CFO Review` task is assigned a value of "0". This path is selected if the expression evaluates to False. The branch path pointing to `Accounting WQ Review` task has a value of "1". This path is taken if the expression evaluates to True. Therefore, the expression must evaluate to True when the `Total` field is less than $10,000.

This section of the tutorial describes how to:

- Configure the second Branch Task

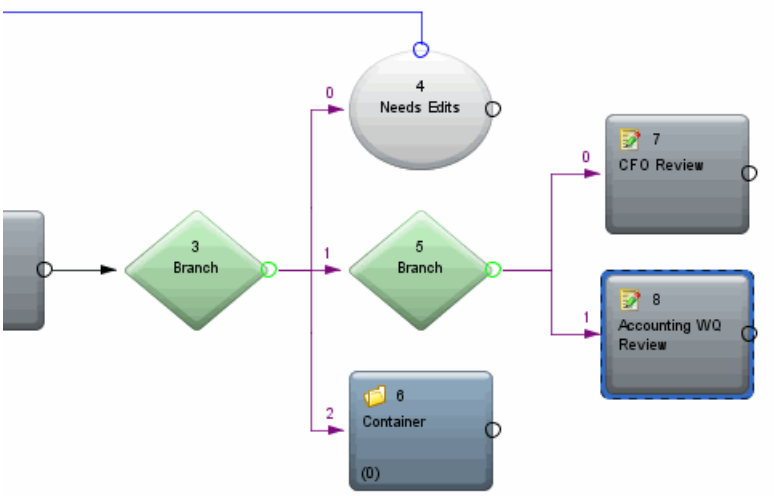
Configure the second Branch Task

Use the following procedure to configure the branch task.

To configure the second Branch task

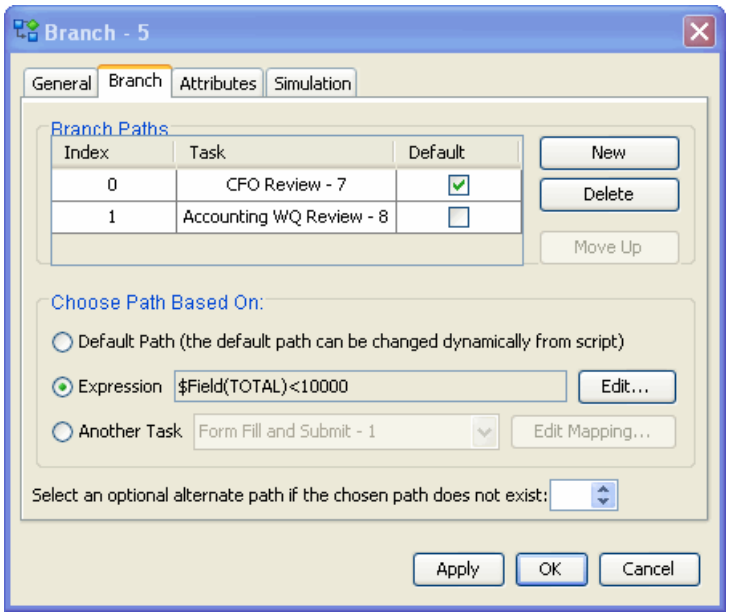
1. Before a Branch task can be configured, you must create its successor tasks. Add two Form tasks to the process template.

2. Connect the second branch task to the Form tasks as shown in the following diagram.



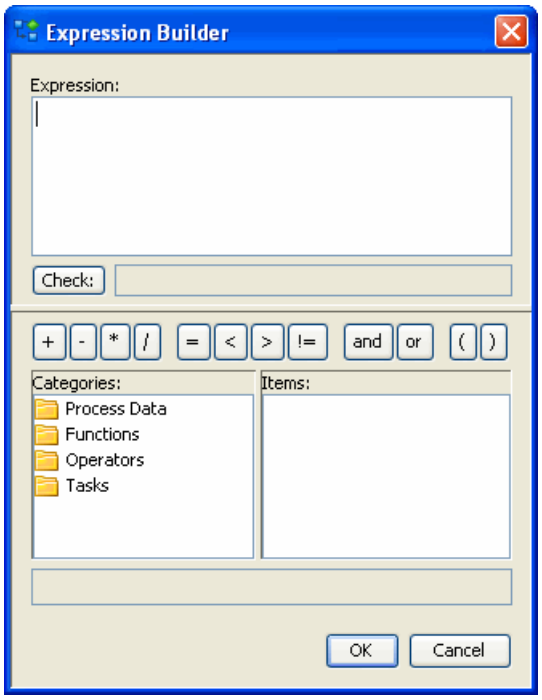
3. Rename the Form tasks.
   a. Double-click the Form task connected to path 0, and rename it to `CFO Review`. Click **OK**.
   b. Double-click Form task connected to path 1, and rename it to `Accounting WQ Review`. Click **OK**.
4. Double-click the second Branch task to edit its properties.
5. In the **Name** box, type **`Approved1`**.

   The Duration is automatically set to `0:00:00:01.000` (1 second).
6. Click the **Branch** tab.



7. In the **Choose Path Based On** area, click **Expression** and then click **Edit**.

   The **Expression Builder** dialog box opens.



8. In the **Categories** list, click the **Process Data** > **Fields** folder.

The fields associated with the `Purchase Order` form are displayed in the **Items** list.

9. In the **Items** list, double-click **Total** .

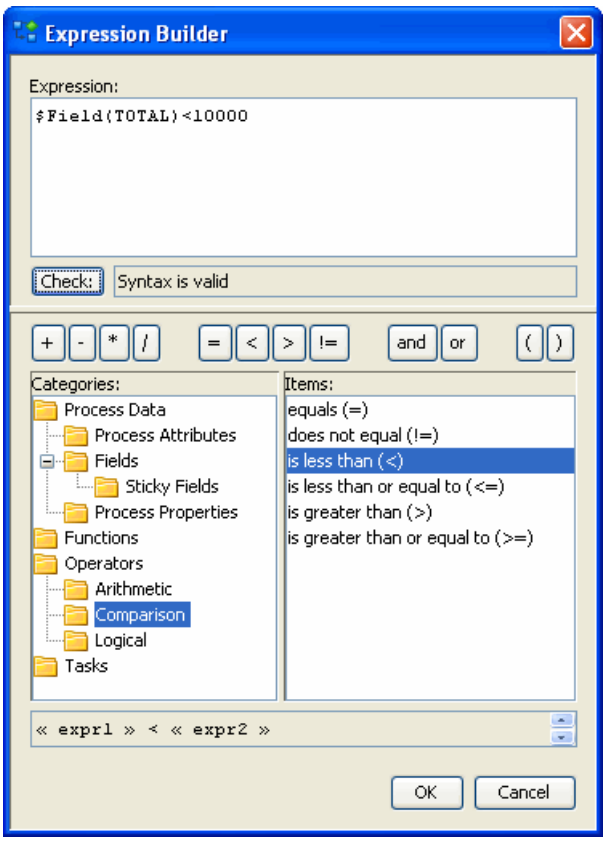
   `$Field(TOTAL)`is added to the **Expression** box.

10. In the **Categories** list, click the **Operators** > **Comparison** folder.
11. In the Items list, double-click **is less than (<)**.

    The operator is added to the **Expression** box.

12. In the **Expression** box, place the cursor after less than character (<) and type **`10000`**.
13. Click **Check**.

    The Expression Builder verifies whether the expression is valid. The message `Syntax is valid` is displayed.



14. Click **OK** to save the expression.

    The Expression is displayed in the branch task properties dialog box.

15. Click **OK** to close the branch task properties dialog box.

16. Save the process.

Related Topics

- “Build Expressions” on page 32
- “Configure Tasks” on page 91
- “Configure General Properties” on page 92
- “Configure a Branch Task” on page 100

Configure the Rejected Path

If the supervisor rejects the `Purchase Order` form, the first branch task (task id: 3) executes the Container Task. The Container task is a container that is used to group other tasks.

In the `Purchase Order` process, the container task holds a sequence of tasks that send a copy of the rejected form to the process originator, and send an e-mail stating that the form was rejected.

This section of the tutorial describes how to:

- Configure the Container Task
- Configure the Form Copy Task
- Configure the E-mail Task using Substitution Syntax

Configure the Container Task

Use this procedure to:

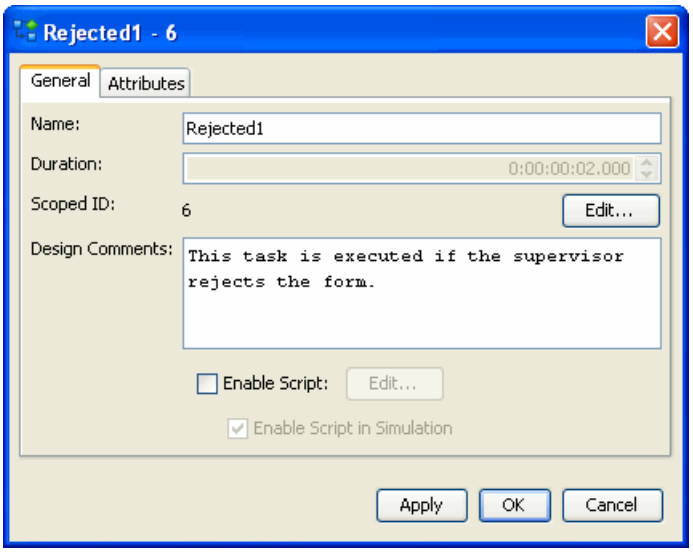
- configure the general properties of the container task.
- open the container task.
- add tasks inside the container task.

To configure the container task

1. Right-click the container task and click **Task Properties**.

   The container task properties dialog box is displayed.

NOTE Double-clicking a Container task does not show the task properties. Instead, the contents of the container are displayed.



2. In the **Name** box, type **`Rejected1`**.

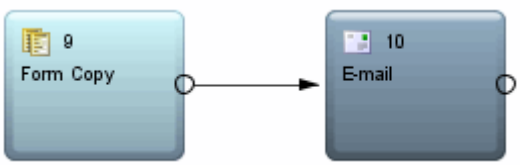
TIP The Duration of a Container task is set automatically. The duration is set to the greatest value of:

- the total durations linked tasks
- the longest Duration of a non-linked tasks.

3. In the **Design Comments** box, type the following text:

   `This task is executed if the supervisor rejects the form.`

4. Click **OK**.
5. Double-click the `Rejected1` (Container) task. The process template now looks blank. This is because you are viewing the empty container.
6. Add a Form Copy task and an Email task to the template.
7. Connect the tasks as shown in the following diagram:



Related Topics

- “Add a Task” on page 81

- “Connect Tasks” on page 83
- “Container Task” on page 72
- “Configure Tasks” on page 91
- “Configure General Properties” on page 92

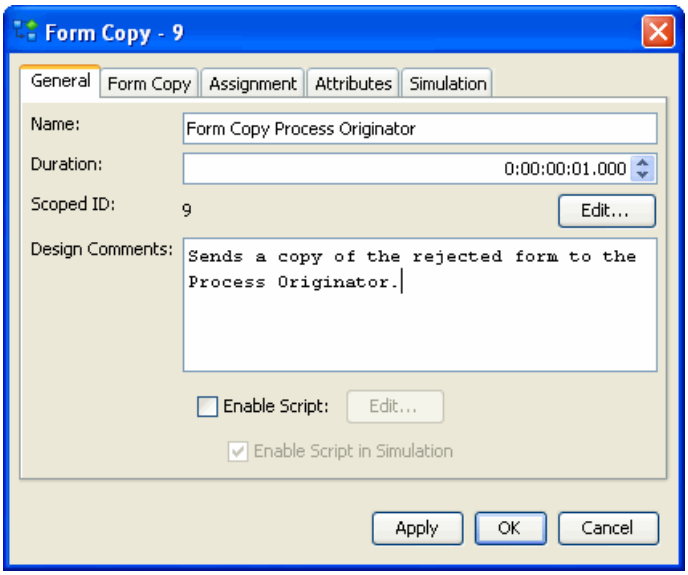
Configure the Form Copy Task

Use this procedure to configure the Form Copy task. This task is used to send a copy of the rejected form to the process originator’s inbox.

To configure the form copy task

1. Double-click the Form Copy task to edit its properties.

   The properties dialog box opens.



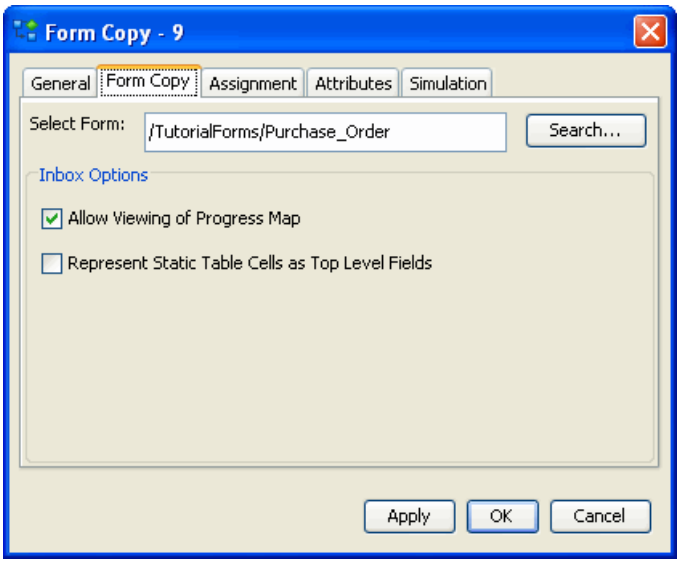
2. In the **Name** box, type **`Form Copy Process Originator.`**

   The duration of the task is set to 1 second by default. As the task is automated, do not change the duration.

3. In the **Design Comments** box, type the following:

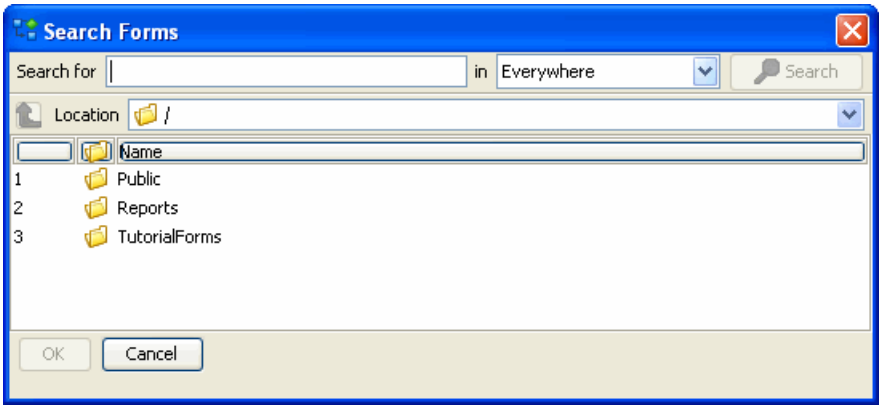
   `Sends a copy of the rejected form to the Process Originator.`

4. Click the **Form Copy** tab.

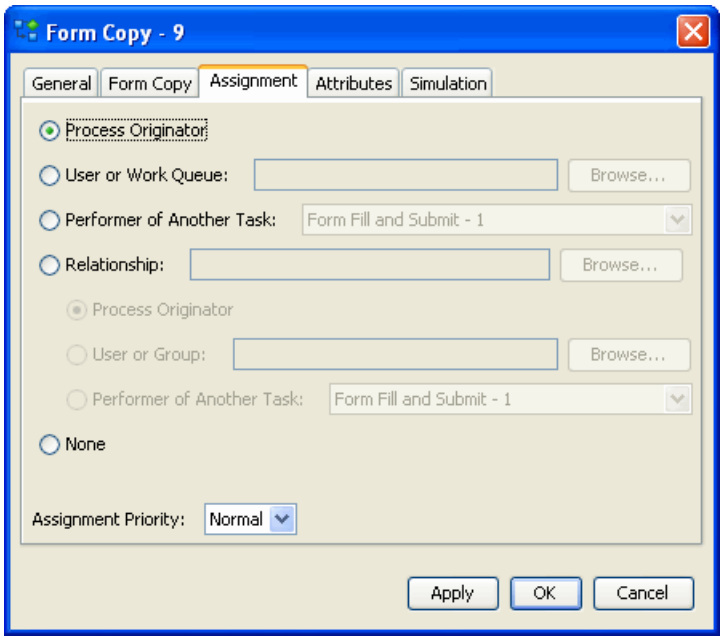


5. Select the **Allow Viewing of Progress Map** check box.
6. Click **Search**.

   The **Search Forms** dialog box opens.



7. Select the `Purchase Order` form in the `TutorialForms` folder, and click **OK**.
8. Click the **Assignment** tab.



9. Click **Process Originator**.
10. Click **OK**.

Related Topics

- “Form Copy Task” on page 77
- “Configure Tasks” on page 91
- “Configure General Properties” on page 92
- “Configure a Form Copy Task” on page 114
- “Task Assignment” on page 95
- “Search the APA Server” on page 132

Configure the E-mail Task using Substitution Syntax

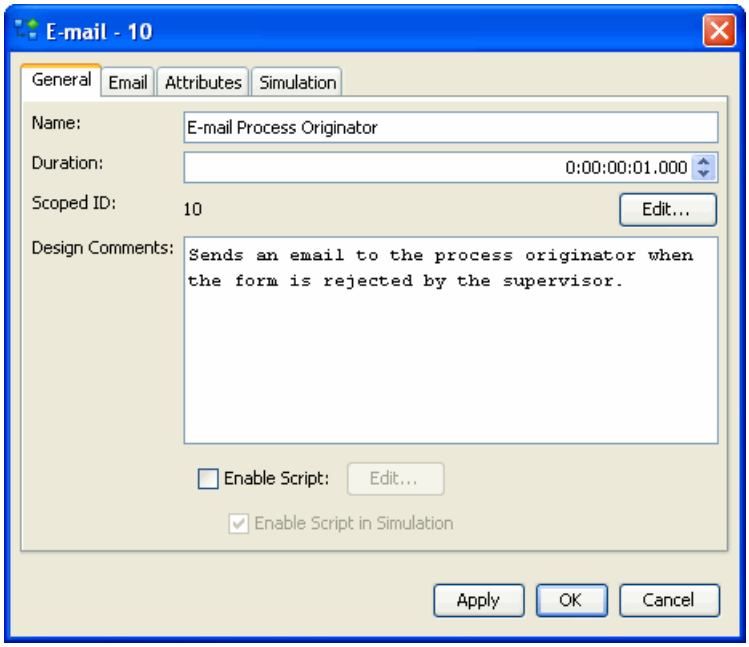
The e-mail task sends an e-mail to the process originator to notify them that the purchase order has been rejected. Substitution syntax is used in the e-mail message so that (regardless of the identity of the process originator), the message is sent to the correct address. Substitution syntax also allows you to display values from the submitted form in the e-mail message.

Use the following procedure to configure the e-mail task.

To configure the E-mail task

1. Double click the Email task to edit its properties.

   The properties dialog box opens.



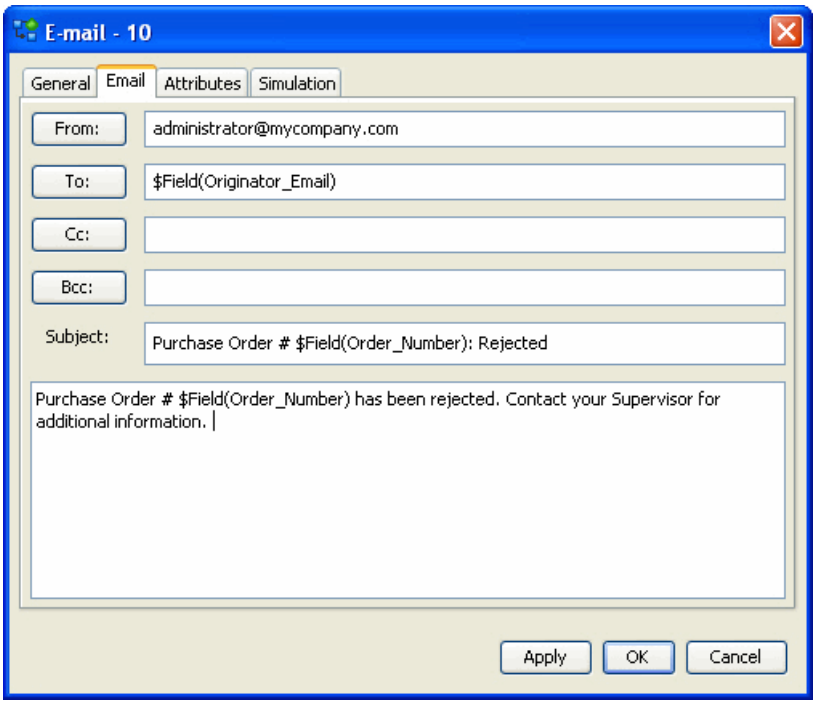
2. In the **Name** box, type **`E-mail Process Originator`**.

   The Duration is set automatically to `0:00:00:01.000` (1 second).

3. In the **Design Comments** box, type the following text:

   `Sends an email to the process originator when the form is rejected by the supervisor`.

4. Click the **Email** tab.



5. In the **From** box, type the e-mail address of your APA Administrator (for example, **`administrator@mycompany.com`**).
6. In the **To** box, type **`$Field(Originator_Email)`**.

   When the e-mail is sent, the substitution syntax `$Field(Originator_Email)` is replaced with the actual e-mail address provided in the `Originator_Email` field on `Purchase_Order` form.
7. In the **Subject** box, type **`Purchase Order # $Field(Order_Number): Rejected`**.

   When the e-mail is sent, the substitution syntax `$Field(Order_Number)`is replaced with the actual purchase order number used on the `Purchase_Order` form.
8. Type the e-mail message:

   `Purchase Order # $Field(Order_Number) has been rejected. Contact your Supervisor for additional information.`
9. Click **OK**.
10. Click **Parent level** to return to the parent process template. Alternatively, double-click **Purchase_Order_Process** in the Process Tree.
11. Save the process.

Related Topics

- "E-mail Task" on page 78
- "Configure Tasks" on page 91
- "Configure General Properties" on page 92
- "Configure an E-mail Task" on page 107
- "Create Dynamic Values Using Substitution Syntax" on page 34

Configure the CFO Review Task

When the value of the purchase order submitted by the process originator is $10,000 or greater (and the originator's supervisor approves the form), the `CFO Review` task is executed. This task is assigned to the Chief Financial Officer (using the `Tutorial_CFO Relationship`). The CFO receives the completed Purchase Order form in their Inbox. The CFO can then review the form and choose to approve it, reject it, or send it back to the originator for editing.

This section of the tutorial describes how to:

- Configure the CFO Review Form Task

Configure the CFO Review Form Task

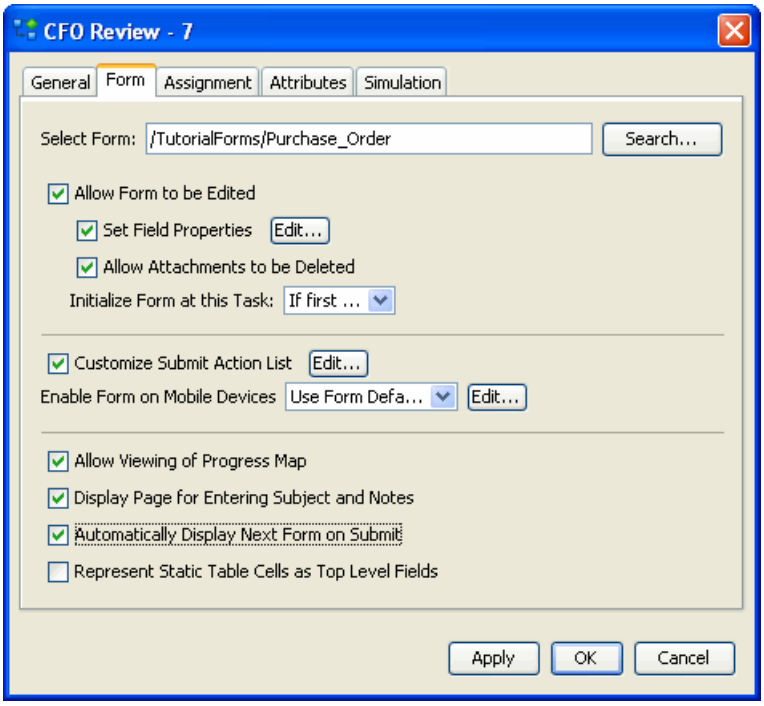
Use the following procedure to:

- configure the general properties of the `CFO Review` form task.
- assign the `Purchase_Order` form to the task
- modify the properties of the form fields so that the CFO cannot edit the form (only sign it)
- customize the submit actions available on the form. Custom submit actions allow the CFO to choose what to do with the form (approve it, reject it, or return it to the originator for editing).
- configure the form to display a notes page on submission. The CFO can then type notes to explain why the form has been rejected or returned for editing.

To configure the CFO Review Form task

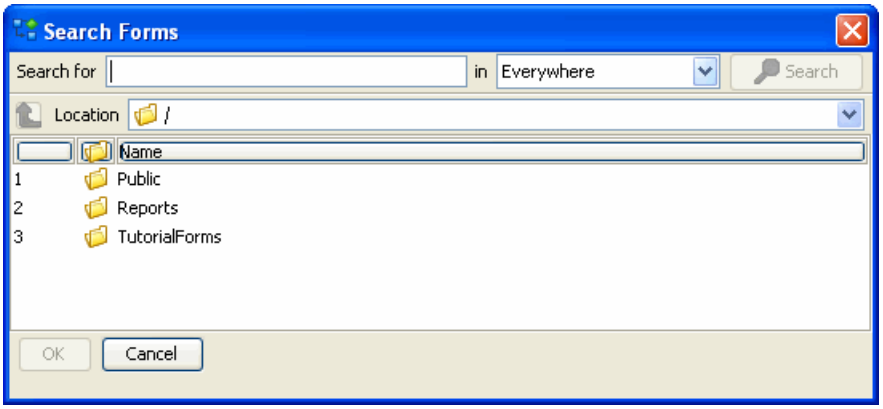
1. Double-click the `CFO Review` task to edit its properties. (The task has already been named).
2. In the **Duration** box, type **`0:00:05:00.000`** (5 minutes).

3. In the **Design Comments** box, type a suitable description to explain the purpose of the task.
4. Click the **Form** tab.



5. Click **Search**.

   The **Search Forms** dialog box is displayed.

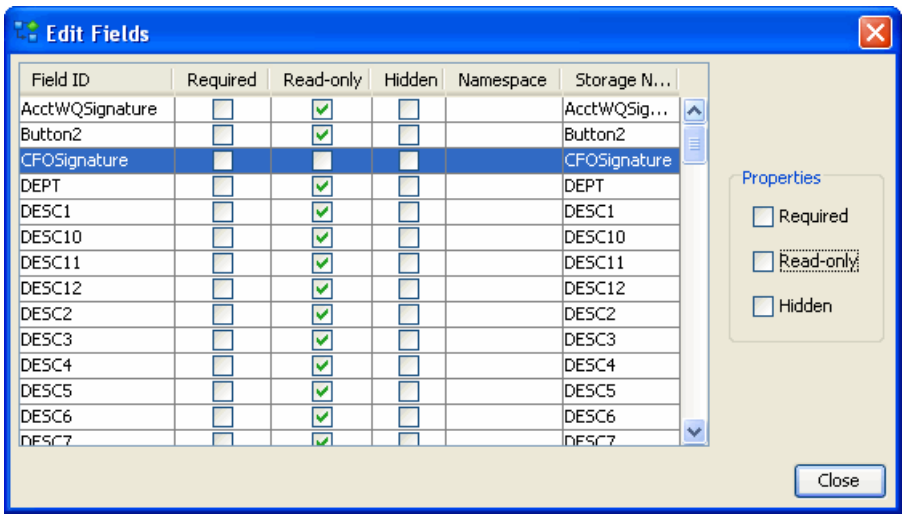


6. Select the `Purchase Order` form in the `TutorialForms` folder, and click **OK**.
7. Select the following check boxes:

- **Allow Form to be Edited**
- **Allow Attachments to be Deleted**
- **Allow Viewing of Progress Map**
- **Display Page for Entering Subject and Notes**
- **Automatically Display Next Form on Submit**.

8. Select the **Set Field Properties** check box and click **Edit**.

   The **Edit Fields** dialog box opens. The list of fields in this dialog box is built dynamically from the APA server.

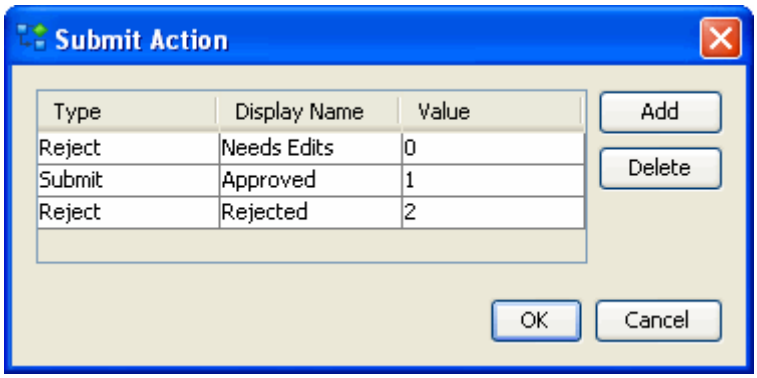


9. Set all of the fields (except `CFOSignature`) to Read Only.

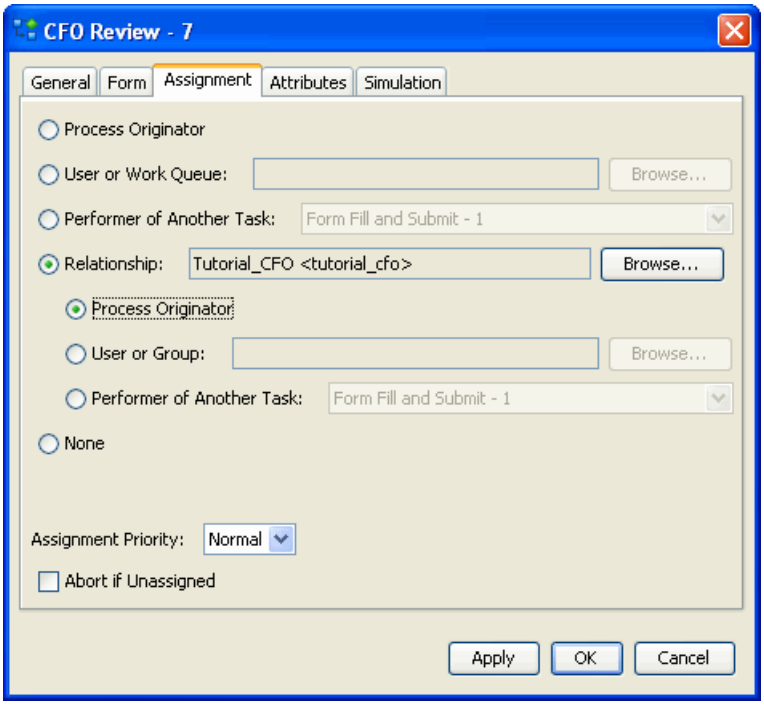
   Select all of the fields, except `CFOSignature`) and select the **Read-only** check box in the **Properties** area.

10. Click **OK**.
11. Select the **Customize Submit Action List** check box and click **Edit**.

    The **Submit Action** dialog box opens.

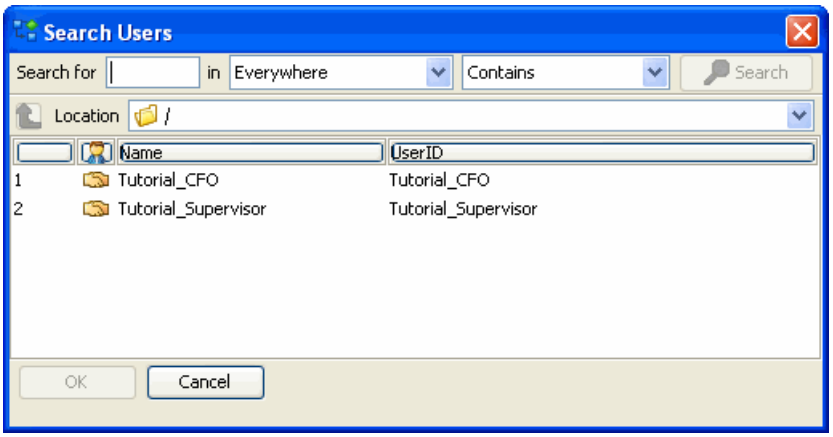


12. Configure the custom submit actions for the form as shown in the preceding image. Click **Add** to add custom submit actions. Double-click a cell to type or select a value.
13. Click **OK**.
14. Click the **Assignment** tab.



15. Click **Relationship** and click **Browse**.

    The **Search Users** dialog box opens.



16. Click **Tutorial_CFO** and click **OK**.
17. Click the **Process Originator** relationship type.
18. Click **OK**.
19. Save the process.

Related Topics

- “Form Task” on page 77
- “Configure Tasks” on page 91
- “Configure General Properties” on page 92
- “Configure a Form Task” on page 113
- “Task Assignment” on page 95
- “Search the APA Server” on page 132

Configure the Accounting Work Queue Review Task

The `Accounting WQ Review` task executes when the CFO approves the purchase order, or if the supervisor approves the purchase order and the total value is not $10,000 or greater.

This task is assigned to the `Tutorial_Accounting` Work Queue. The Work Queue receives the completed Purchase Order form in the Work Queue Inbox. An individual in the `Tutorial_Accounting` Work Queue reviews the form and then choose whether to approve the form, reject it, or send it back to the originator for editing.

This section of the tutorial describes how to:

- Configure the Accounting WQ Review Form Task

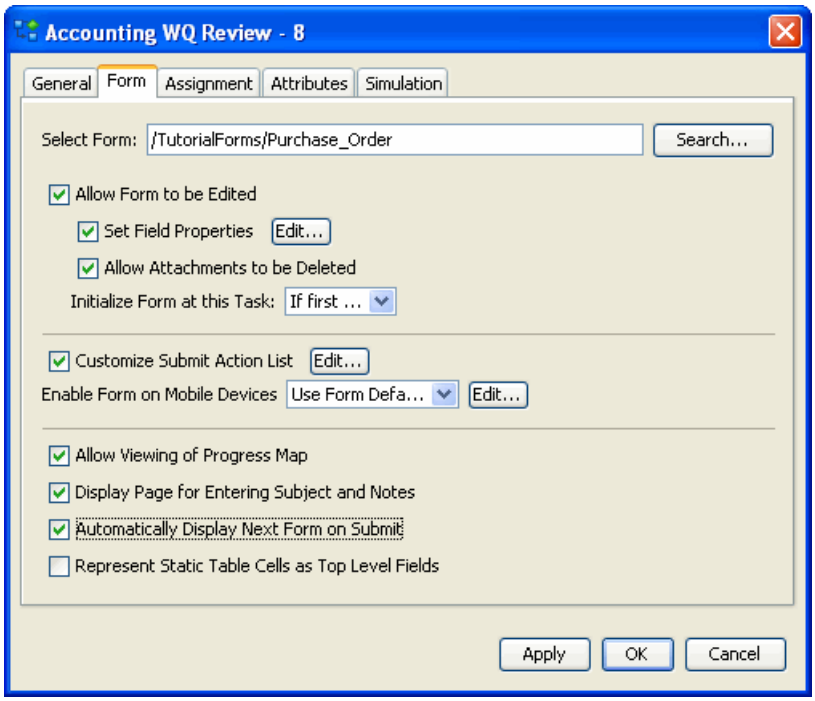
Configure the Accounting WQ Review Form Task

Use this procedure to:

- configure the properties of the `Accounting WQ Review` Form Task
- assign the task to a work queue

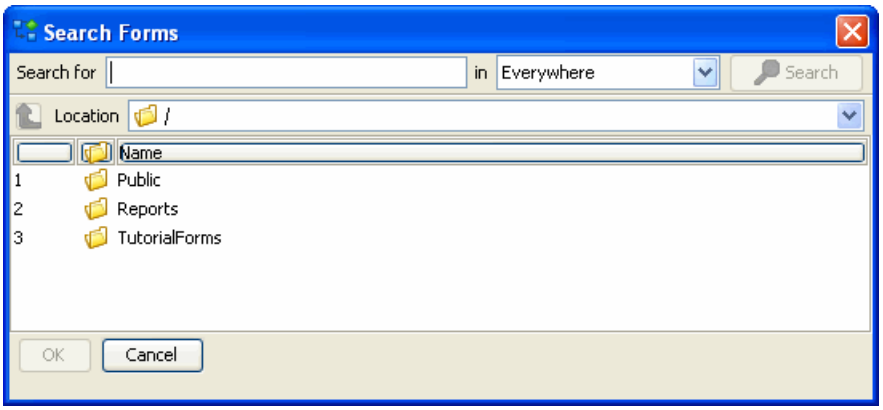
To assign the Accounting WQ Review task to a work queue

1. Double-click the `Accounting WQ Review` task to edit its properties. The task has already been named.
2. In the **Duration** box, type **`0:00:05:00.000`** (5 minutes).
3. In the **Design Comments** box, type a suitable description for the task.
4. Click the **Form** tab.



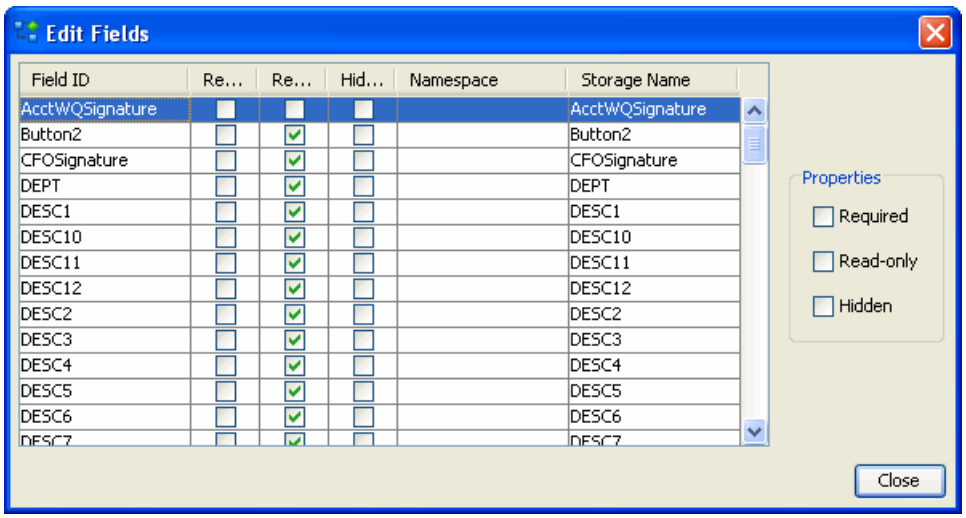
5. Click **Search**.

   The **Search Forms** dialog box is displayed.



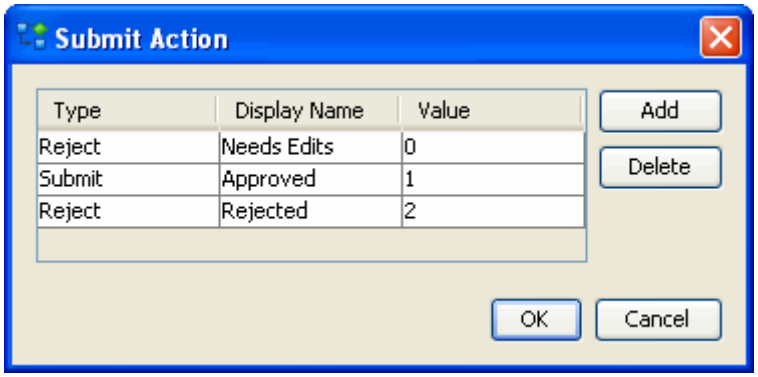
6. Select the `Purchase Order` form in the `TutorialForms` folder, and click **OK**.
7. Select the following check boxes:
   - **Allow Form to be Edited**
   - **Allow Attachments to be Deleted**
   - **Allow Viewing of Progress Map**
   - **Display Page for Entering Subject and Notes**
   - **Automatically Display Next Form on Submit**
8. Select the **Set Field Properties** check box and click **Edit**.

   The **Edit Fields** dialog box opens. The list of fields is built dynamically by the APA server.

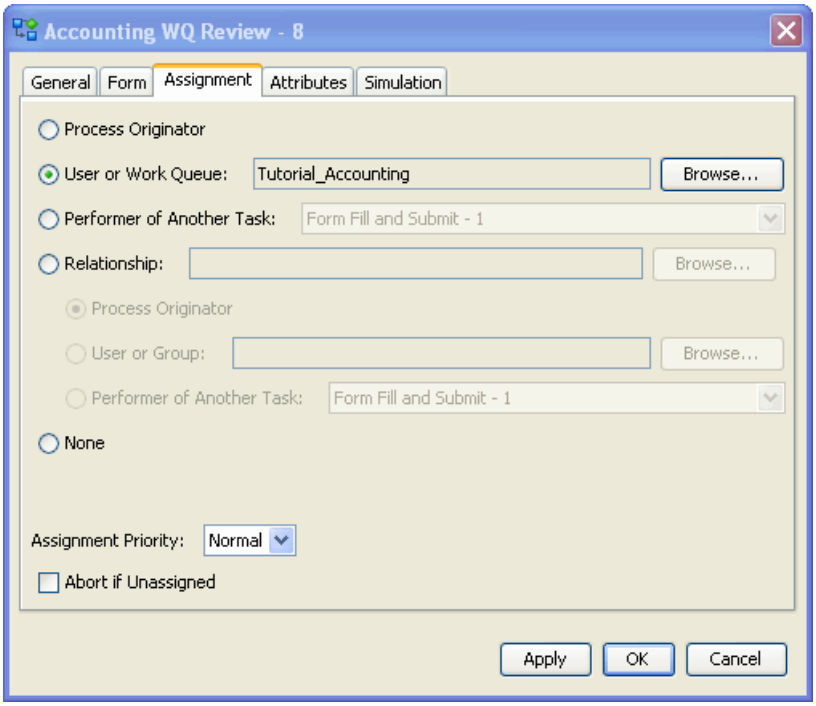


9. Set all of the fields (except `AcctWQSignature`) to Read Only. To do this, highlight the fields in the table, and select the **Read-only** check box.
10. Click **OK**.
11. Select the **Customize Submit Action List** check box and click **Edit**.

    The **Submit Action** dialog box opens.

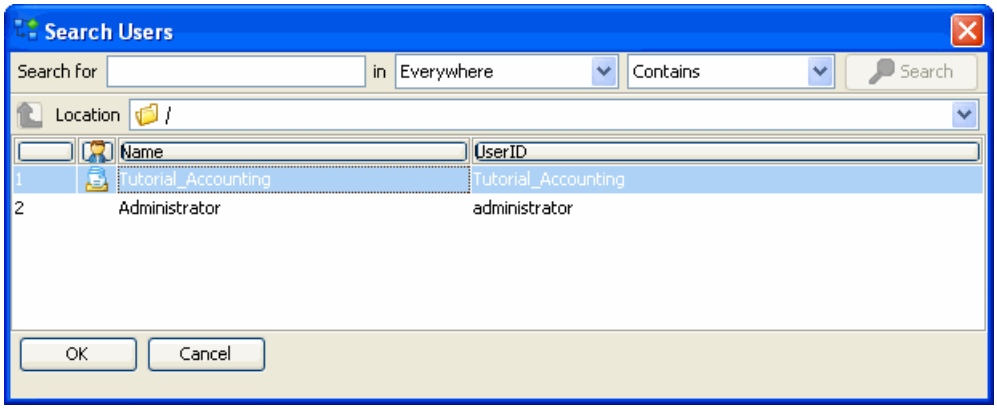


12. Configure the custom submit actions for the form as shown in the preceding image. Click **Add** to add custom submit actions. Double-click a cell to type or select a value.
13. Click **OK**.
14. Click the **Assignment** tab.



15. Click **User or Work Queue** and click **Browse**.

    The **Search Users** dialog box opens.



16. Click the **Tutorial Accounting** Work Queue and click **OK**.
17. Click **OK**.
18. Save the process.

Related Topics

- "Form Task" on page 77

- "Configure Tasks" on page 91
- "Configure General Properties" on page 92
- "Configure a Form Task" on page 113
- "Task Assignment" on page 95
- "Search the APA Server" on page 132

Configure the CFO Review Branch Task

The next step in creating the purchase order process is to configure the tasks that follow the `CFO Review` task. The `CFO Review` task is followed by a branch task. The branch task executes a process path based on the submit action (approve, reject, send for edit) selected by the CFO.

This section of the tutorial describes how to:

- Add and Connect the CFO Review Branch Task
- Configure the CFO Review Branch Task

Add and Connect the CFO Review Branch Task

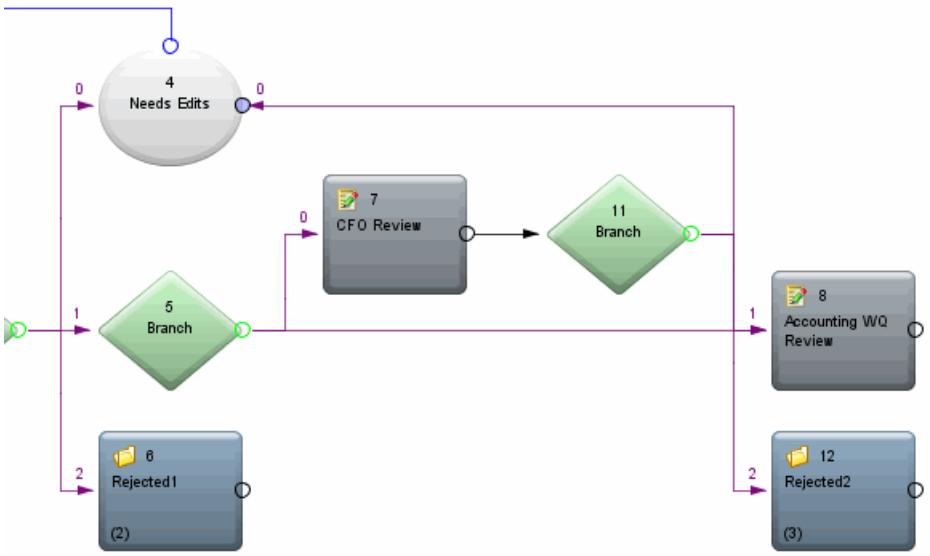
Use the following procedure to add the branch task to the process and connect it to the other tasks.

To add and connect the branch task

1. Add a Branch task to the process template.
2. Connect the `CFO Review` task to the Branch task.
3. Add a new container task to the process template. Name the container **`Rejected2`**.
4. Connect the Branch task to its successor tasks.
   a. Connect the Branch task to the `Needs Edits` (loop) task.
   b. Connect the Branch task to the `Accounting WQ Review` (form) task.
   c. Connect the Branch task to the new container task.

TIP You might need to move some of the tasks on the template to see the paths clearly.

The end of the process template should now look similar to the following diagram:



Related Topics

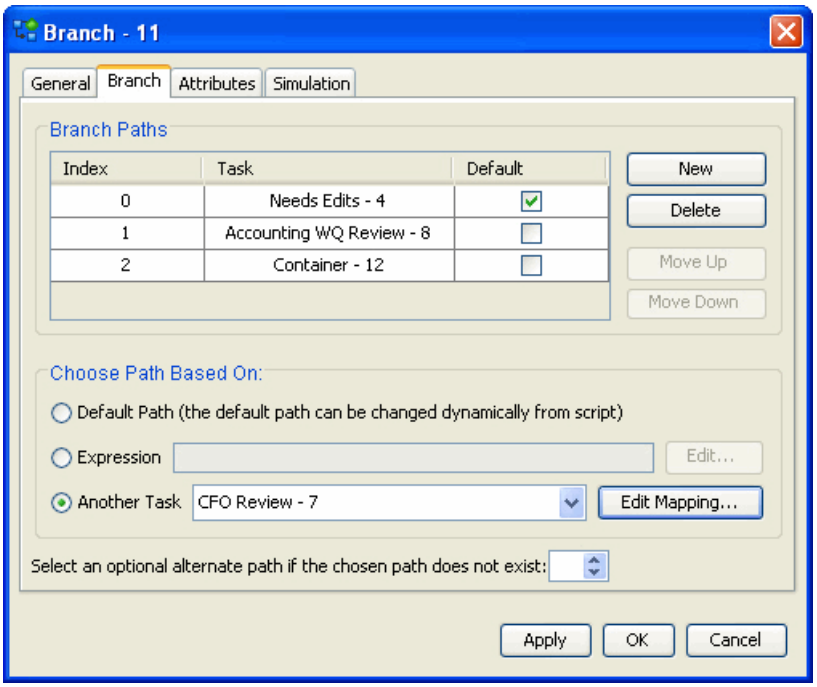
- “Branch Task” on page 74
- “Add a Task” on page 81
- “Connect a Branch Task” on page 84

Configure the CFO Review Branch Task

Use the following procedure to configure the branch task.

To configure the branch task

1. Double-click the Branch task to edit its properties.
2. Click the **Branch** tab.

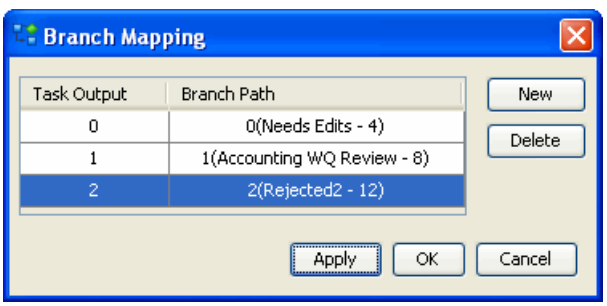


3. In the **Branch Paths** area, the `Needs Edits`, `Accounting WQ Review`, and `Container` tasks have been assigned Index values. If the path Index values for your branch task do not match those shown in the preceding image, select a row in the **Branch Paths** table and click **Move Up** or **Move Down**.

   The path index for the `Needs Edits` task has a value of 0. The path index for the `Accounting WQ Review` task has a value of 1. The path index for the Container task has a value of 2.

4. The Branch task should choose a path based on the submit action chosen by the CFO. In the **Choose Path Based On** area, click **Another Task**.
5. In the **Another Task** list, click the **CFO Review** task.
6. Click **Edit Mapping**.

   The **Branch Mapping** dialog box opens.



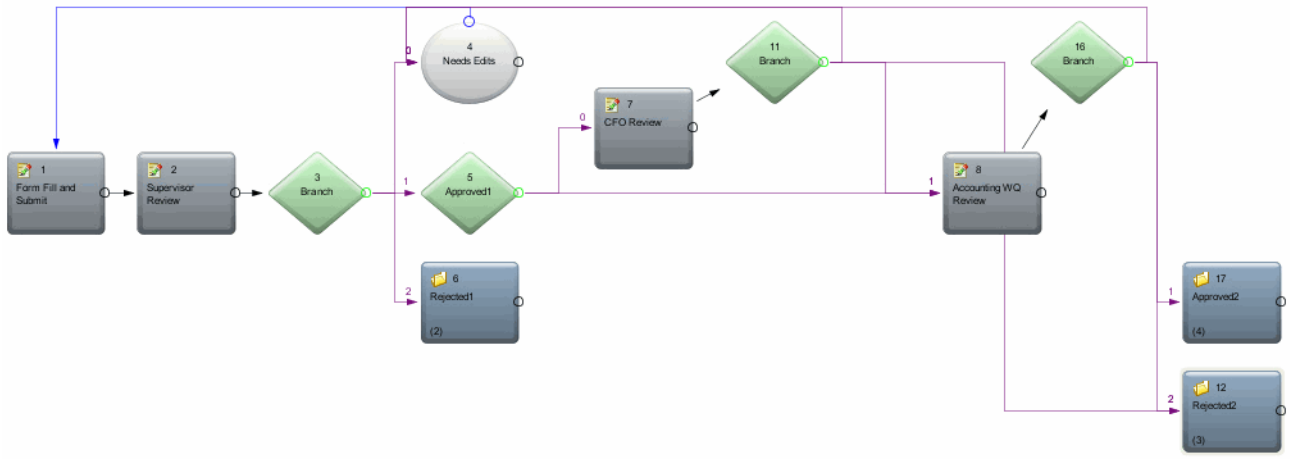
7. Click **New** three times to add new branch mapping entries to the list.
8. Configure the **Task Output** and **Branch Path** settings by double-clicking each cell in the table and clicking a task output or branch paths from the list.
   - Task Output `0` should map to Branch Path `0` (the `Needs Edits` loop).
   - Task Output `1` should map to Branch Path `1` (the `Accounting WQ Review` task).
   - Task Output 2 should map to Branch Path `2` (the `Rejected2` container).
9. Click **OK**.
10. Click **OK** again.
11. Save the process.

Related Topics

- "Configure Tasks" on page 91
- "Configure a Branch Task" on page 100

Configure the Accounting Review Branch Task

A branch task is used following the `Accounting WQ Review` task to select the process path based on the result of the review. If the form is approved by the Accounting Work Queue, the `Approved` container task is executed. If the form is rejected, the `Rejected2` container task executes. If the form is submitted for further editing, the `Needs Edits` loop task executes.



This section of the tutorial describes how to:

- Add and Connect the Accounting Review Branch Task
- Configure the Accounting Review Branch Task

Add and Connect the Accounting Review Branch Task

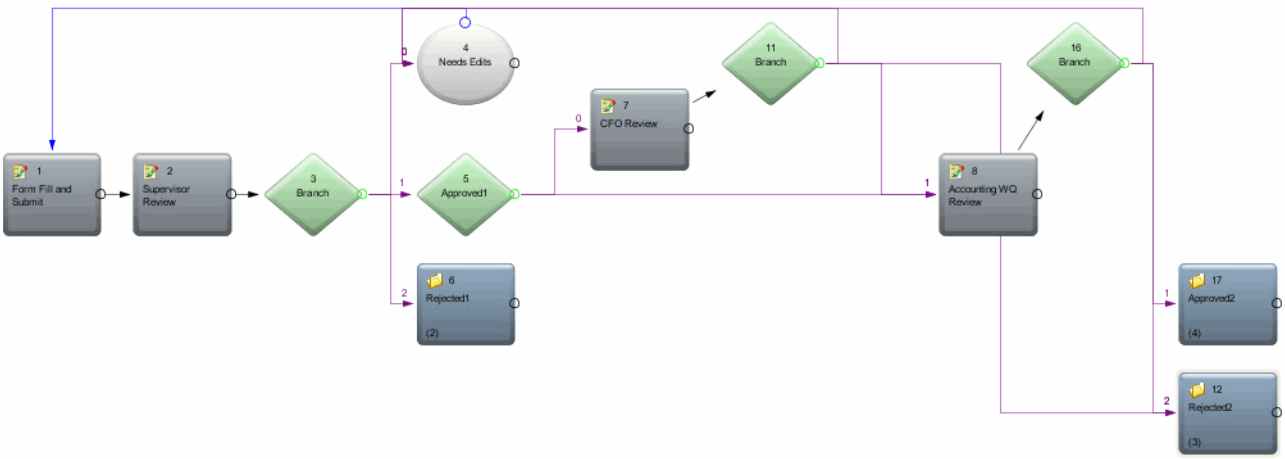
Use this procedure to add the `Accounting WQ Review` branch task.

To add and connect the branch task

1. Add a Branch task to the template.
2. Connect the `Accounting WQ Review` task to the Branch task.
3. Add a Container task to the process template. Name the container **`Approved`**.
4. Connect the Branch task to its successor tasks.
   a. Connect the Branch task to the `Needs Edits` (Loop) task.
   b. Connect the Branch task to the `Approved` (Container) task.
   c. Connect the Branch task to the `Rejected2` (Container) task.

TIP You might need to move some of the tasks on the template to see the paths clearly.

The process should now look similar to the following diagram.



Related Topics

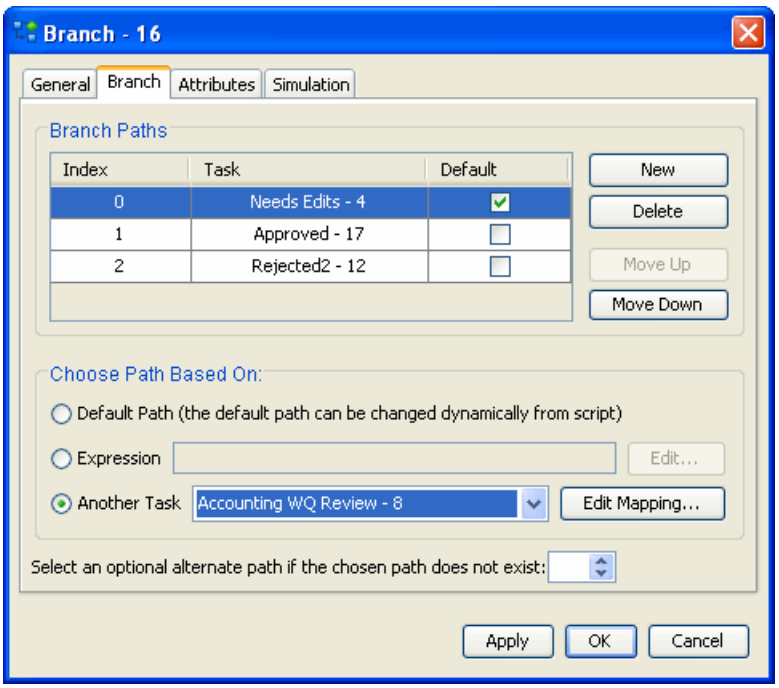
- “Branch Task” on page 74
- “Add a Task” on page 81
- “Connect a Branch Task” on page 84

Configure the Accounting Review Branch Task

Use this procedure to configure the `Accounting WQ Review` branch task.

To configure the branch task

1. Double-click the Branch task to edit its properties.
2. Click the **Branch** tab.



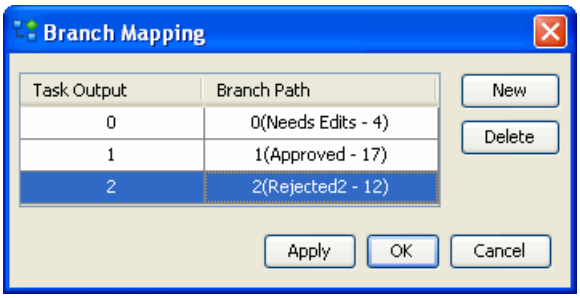
3. In the **Branch Paths** area, the `Needs Edits`, `Approved`, and `Rejected2` tasks are assigned Index values. These values are used to map the Branch paths to the `Accounting WQ review` task output.

   If the path Index values for your branch task do not match those shown in the preceding image, select a row in the **Branch Paths** table and click **Move Up** or **Move Down**.

4. The Branch task should choose a path based on the submit action chosen by the CFO. In the **Choose Path Based On** area, click **Another Task**.

5. In the **Another Task** list, click the **CFO Review** task.
6. Click **Edit Mapping**.

   The **Branch Mapping** dialog box opens.



7. Click **New** three times to add new branch mapping entries to the list.
8. Configure the **Task Output** and **Branch Path** settings by double-clicking each cell in the table and clicking a task output or branch path from the list.
   - Task Output `0` should map to Branch Path `0` (the `Needs Edits` loop).
   - Task Output `1` should map to Branch Path `1` (the `Approved` container).
   - Task Output 2 should map to Branch Path `2` (the `Rejected2` container).
9. Click **OK**.
10. Click **OK** again.
11. Save the process.

Related Topics

- “Configure Tasks” on page 91
- “Configure a Branch Task” on page 100

Configure the Rejected2 Container

If the CFO or Accounting Work Queue reject the purchase order form, the `Rejected2` container is executed.

The `Rejected2` Container is used to send a copy of the rejected form to the process originator and their supervisor. The container also contains an e-mail task to send the process originator an e-mail stating that the form was rejected.

This section of the tutorial describes how to:

- Configure the Rejected2 Container Task

- Add Tasks to the Container
- Send a Copy of the Form to the Supervisor

Configure the Rejected2 Container Task

Use the following procedure to configure the general properties of the container task.

To configure the container task

1. Right-click the Container task and click **Task Properties**.

TIP The Duration of a Container task is set automatically. The duration is set to the greatest value of:

- the total durations linked tasks
- the longest Duration of a non-linked tasks.

2. In the **Design Comments** box, type the following text:

   ```
   This task is executed if the CFO or Accounting Work Queue
   reject the Purchase Order form.
   ```

3. Click **OK**.

Related Topics

- “Configure Tasks” on page 91
- “Configure General Properties” on page 92

Add Tasks to the Container

The Rejected2 container task performs a similar function to the Rejected1 container. To save time, the tasks from the Rejected1 container can be copied and modified.

Use the following procedure to add tasks to the Rejected2 container.

To add tasks to the Rejected2 container

1. Double-click the `Rejected1` (Container) task in the **Process Tree**. This is the Container task that was configured earlier in the tutorial.
2. Select the tasks in the graphical view using one of the following methods:
   - On the **Edit** menu, click **Select All**.
   - Drag a bounding outline around the tasks

- Click the first task, hold down SHIFT, and click the second task.

3. Copy the tasks.
   - On the **Edit** menu, click **Copy**.
   - Right-click the selected tasks and click **Copy**.
4. Double click the `Rejected2` (container) task in the **Process Tree** to open the second container task.

   The **Graphical View** is blank because no tasks have been added to the container.
5. Right-click in the **Graphical View** and click **Paste**.
6. The form copy task has already been configured to send a copy of the `Purchase_Order` form to the Process Originator (see "Configure the Form Copy Task" on page 218)
7. Double click the Email task to edit its properties. As the previous email task was used when the form was rejected by the Process Originator's supervisor (and this task is intended for a rejection by the CFO or Accounting Work Queue), make the following changes:
   - In the **Design comments** box (in the **General** tab), type:

     ```
     Sends an email to the process originator when the form
     is rejected by the CFO or Accounting Work Queue.
     ```
   - In the body of the e-mail message (in the **Email** tab), type:

     ```
     Purchase Order # $Field(Order_Number) has been
     rejected by the CFO or Accounting Work Queue.
     ```
8. Click **OK**.

Related Topics

- "Configure Tasks" on page 91
- "Configure General Properties" on page 92
- "Configure an E-mail Task" on page 107
- "Create Dynamic Values Using Substitution Syntax" on page 34

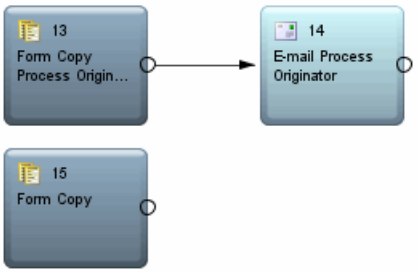
Send a Copy of the Form to the Supervisor

A form copy task must be added to the container to send a copy of the rejected form to the originator's supervisor.

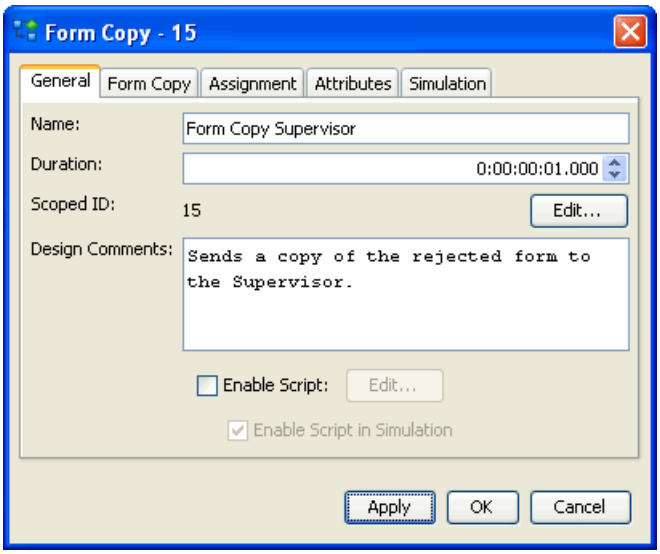
This form copy task can be executed in parallel with the form copy task that sends a copy of the form to the originator.

To send a copy of the form to the supervisor

1. Add another Form Copy task to the process template. The process template should look similar to the following diagram:



2. Double-click the new Form Copy task to edit its properties.



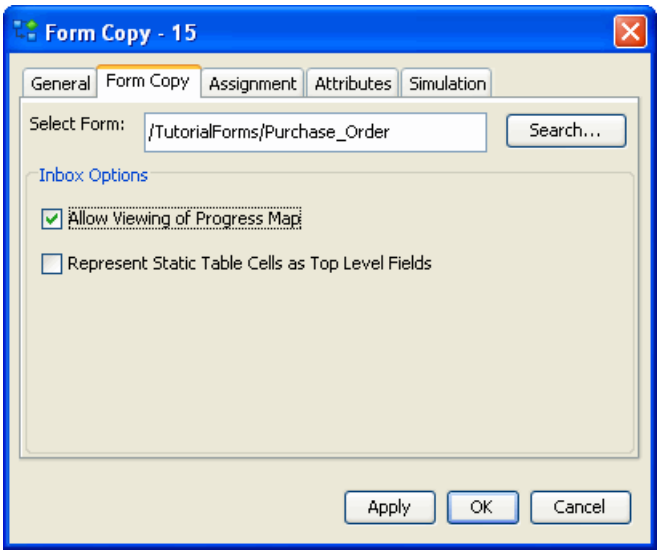
3. In the **Name** box, type **`Form Copy Supervisor`**.

   The duration of the task is set to 1 second by default. As the task is automated, do not change the duration.

4. In the **Design Comments** box, type the following text:

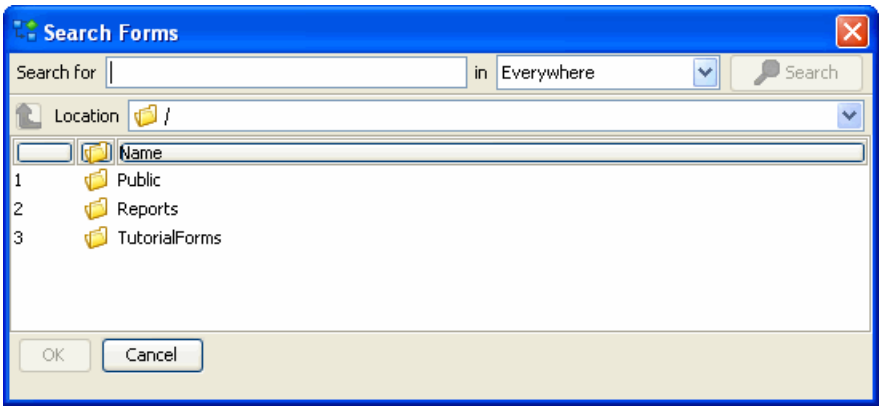
   `Sends a copy of the rejected form to the Supervisor`.

5. Click the **Form Copy** tab.

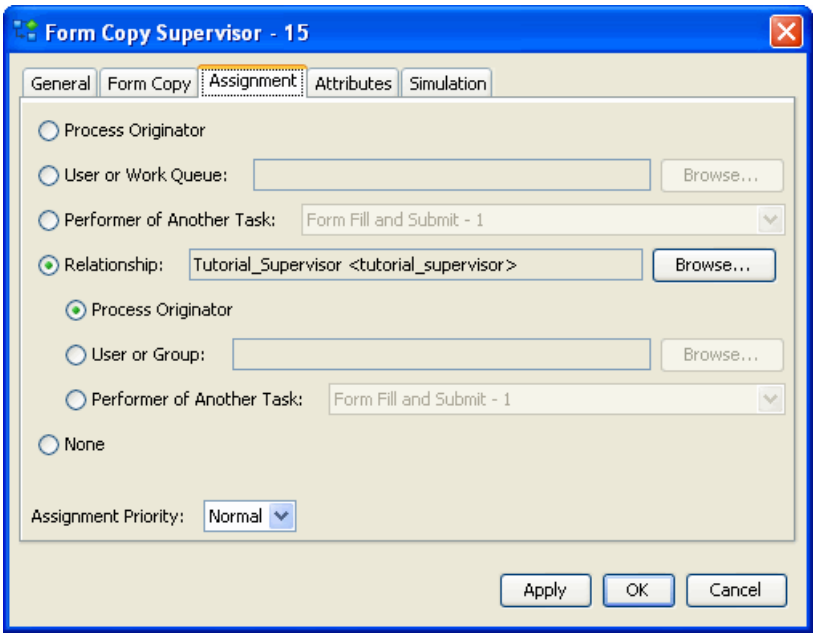


6. Click **Search**.

   The **Search Forms** dialog box opens.

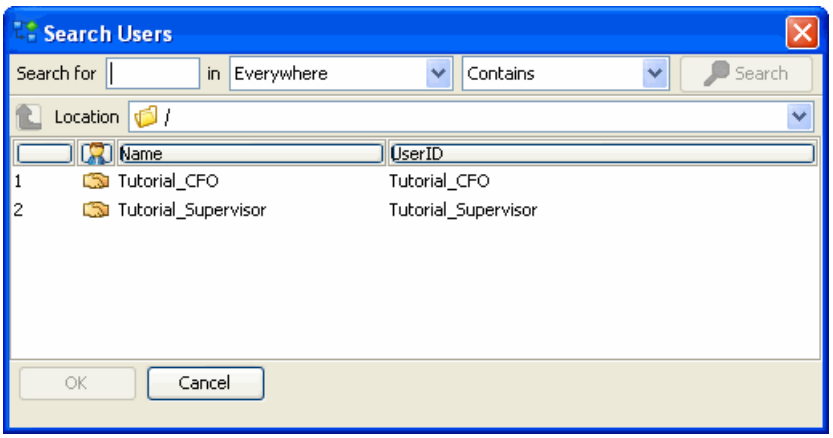


7. Click the `Purchase Order` form in the `TutorialForms` folder, and click **OK**.
8. Select the **Allow Viewing of Progress Map** check box.
9. Click the **Assignment** tab.



10. Click **Relationship.**
11. Click the **Process Originator** relationship option.
12. Click **Browse**.

    The **Search Users** dialog box opens.



13. Click the `Tutorial_Supervisor` Relationship and click **OK**.
14. Click **OK**.
15. Click **Parent level** to return to the parent process template. Alternatively, double-click **Purchase_Order_Process** in the Process Tree.

16. Save the process.

Related Topics

- “Form Copy Task” on page 77
- “Configure General Properties” on page 92
- “Configure a Form Copy Task” on page 114
- “Task Assignment” on page 95
- “Search the APA Server” on page 132

Configure the Approved Container

The `Approved` container task is executed if the Accounting Work Queue approves the purchase order. This is the final approval that is required.

This section of the tutorial describes how to:

- Configure the Approved Container

Configure the Approved Container

The `Approved` Container includes tasks to inform the process originator, the process originator’s supervisor, and the accounting work queue that the form has been approved.

Use the following procedure to configure the container.

To Configure the Approved Container

1. Configure three Form copy tasks to send a copy of the approved form to the Process Originator, the Supervisor, and the Accounting Work Queue.
2. Configure an Email task to e-mail the Process originator and inform them that the Purchase Order has been approved.

   Earlier steps in this tutorial explain how to configure these tasks. For example, see “Configure the Rejected Path” on page 216 or “Configure the Rejected2 Container” on page 238.

The `Purchase Order` Process template is complete. You can now run a Flow Simulation in Process Studio (see “Simulate a Process” on page 123) or publish the process to the APA Server (see “Publish a Process” on page 135).

Index

A

add tasks to process 81
allow attachments to be deleted property 118
allow form to be edited property 117
allow viewing of progress map property 121
and task 72
APA
    server 23
APA Server
    connect 131
    disconnect 132
    retrieve process 47
    search 132
assign tasks
    assignment overview 95
    priority 96
automatically display next form property 121

B

boolean data type 175
branch task 74
business calendar 117

C

change type command 88
close process template 50
command task 78
comments 89
committee task 79
compact tasks 31
composite task 79
    save 87
configure process properties 51
configure tasks 91
connect tasks
    branch task 84
    escalate task 86
    exception task 85
    loop task 85
    overview 83
connect to APA Server 131
console 31
container task 72
create dynamic values 34
create process template 46
create tasks 81
customize submit action list property 118

D

date data type 174
default path (branch task) 74
delay task 75
delete instance 124
delete tasks 88
display page for notes property 121
duplicate instance 124
duration property 93
dynamic values 34

E

e-mail task 78
enable form on mobile devices property 119
escalate task 75
exception task 76
exit command 32
expressions
    data types 172

expression builder 32
field references 175
functions 180
operators 177
wildcard references 176

F

flow simulation 123
form copy task 77
form export task 77
form packet task 77
form task 77

G

gate task 75
graphical view 27
grid lines 31

I

initiate a process 146
instance
delete 124
duplicate 124
overview 27

L

load simulation 127
loop task 74

M

manual task 78
message task 78

N

new window command 30
number data type 173

O

open gate 75
open process template 46
or task 73

P

pack command 31
permissions (publishing) 141
predecessor task 70
print process template 49
process fields 54
process originator 95
process properties
attributes 53
general properties 52
overview 51
process fields 54
process simulation
see simulation
Process Studio
customize 28
process template
add comments 89
close 50
create 46
layout tasks 31
open 46
overview 27
print 49
publish 135
retrieve from APA server 47
save 48
save as image 49
save copy 48
process tree 26
publish process
overview 135
permissions 141
process settings 145

R

redo 31
relationship 95

S

save as image 49
save composite task 87
save process template 48
scoped ID property 93
script task 77
scripts
    default scripting language 28
    insert function 40
    insert task reference 40
    script editor 37
search APA Server 132
servers
    APA 23
simulation
    configure tasks 127
    create flow simulation 123
    create load simulation 129
    delete instance 124
    duplicate instance 124
    flow simulation 123
    flow simulation commands 125
    load simulation 127
    overview 123
    task states 125
sticky note task 80
string concatenation 180
string data type 173
sub process task 72
submit action list 118
substitution syntax 34
successor task 70

T

task library
    customize 28
    overview 26
task patterns 151
task states 125
task types
    and 72
    branch 74
    command 78
    committee 79
    composite 79
    container 72
    delay 75
    e-mail 78
    escalate 75
    exception 76
    form 77
    form copy 77
    form export 77
    form packet 77
    gate 75
    loop 74
    manual 78
    message 78
    or 73
    script 77
    sticky note 80
    sub process 72
    web service 79
tasks
    add to process 81
    assignment 95
    change type 88
    configuration 91
    connect 83
    delete 88
    duration 93
    manual and automated 70
    overview 69
    predecessor and successor 70
    scoped ID 93
template
    see process template
test process
    see simulation

time data type 174
tooltips 28

U

undo 31
use business calendar 117
user interface overview 26

V

views
    graphical 27
    xml 27

W

web service task 79
wildcard references 176
work queue 95

X

xml view 27

Z

zoom controls 30